1996

Index compressed vector quantisation

Jamshid Shanbehzadeh
University of Wollongong

Index Compressed Vector Quantisation

Submitted in fulfilment of the
requirement for the award of the degree

DOCTOR OF PHILOSOPHY

from

UNIVERSITY OF WOLLONGONG

BY

JAMSHID SHANBEHZADEH
(M. E. UNIVERSITY OF TEHERAN 1987)

ELECTRICAL AND COMPUTER ENGINEERING

MARCH 1996

Abstract

Vector quantisation (VQ) is a lossy data compression scheme that has gained tremendous attention because of its simplicity and firm information-theoretic basis. This thesis introduces two groups of novel techniques to improve the performance of traditional VQ (basic VQ) schemes, such as full search or tree-structured, at low bit rates. These new schemes have a simpler implementation, more computational efficiency, and lower memory requirement than other VQ schemes for low bit rate applications. The proposed schemes are capable of providing significant coding improvement over traditional VQ coders, and better results than predictive VQ, finite state VQ, and JPEG at low bit rates.

The aim of the first group of schemes, consisting of three techniques, is to improve the objective quality of basic VQ. These schemes are a combination of traditional VQ coders and a lossless coding technique which compresses the indices (index compressed VQ (IC-VQ)). The performance improvement of these techniques comes from exploiting the indices' correlation. The first index compressed VQ scheme (IC-VQ (I)) exploits the probability of identically indexed neighbouring blocks, and results in an improvement of about 3.5 dB on average over basic VQ. The second index compressed VQ scheme (IC-VQ (II)) addresses the negative effect that an increase in codebook size has on the coding performance of IC-VQ (I). This method employs the fact that the neighbouring blocks' indices are mapped onto a small subset of the codebook. IC-VQ (II) achieves an improvement of about 0.5 dB over IC-VQ (I). The third index compressed VQ scheme (IC-VQ (III)) utilises a variable bit allocation for the indices. The genealogical characteristics has been used in a variable bit allocation. IC-VQ (III) gives an improvement of about 0.7 dB over IC-VQ (II).

The aim of the second group of schemes, consisting of two techniques, is to improve the subjective quality of basic VQ. These methods are a combination of the first group of techniques and a method of improving the subjective quality of basic VQ. The first scheme deals with the problem of using a stationary codebook for non-stationary sources in representing the active areas at low bit rates. The combination of non-stationary codebooks with IC-VQ schemes can provide a solution to this problem. The new method achieves an improvement of about 4 dB over the basic non-stationary codebook-based VQ scheme. The second scheme addresses the problem of disregarding image block activity in bit allocation. The solution proposed is acheived by combining index compression schemes with variable rate multi-stage VQ. This method is computationaly efficient and capable of providing a better coding performance, both objectively and subjectively, than the first method.

Acknowledgements

There are many people who deserve a tremendous amount of thanks and credit for their help during the preparation of this thesis. To write about all these people and their contributions to this work would necessitate another thesis. A few people deserve a special mention. Thanks to my supervisor Dr. Philip Ogunbona. His advice, support, encouragement, friendly manner and time devoted for technical interaction made this work possible.

The assistance of the Department of Electrical and Computer Engineering Staff of the University of Wollongong is acknowledged. There are some people who helped me by offering their experiences; Professor R. M. Gray (Stanford University), Professor E. A. Riskin (University of Washington), and Professor N. M. Nasrabadi (Army Research Lab). I would like to thank them. I also thank Dr. A. Qureshi, P. Vial, and M. Kahani of the Department of Electrical and Computer Engineering for their assistance in proof reading the thesis.

Dedications

I thank God for providing me with the opportunity to carry out this research. This work is dedicated to the following people:

To the people of I. R. of Iran for their financial support.

To my father, mother, my wife (Nayereh) and my children for their patience. Without their support and encouragement this work would not have been completed.

Table of contents

List of notations and Abbreviations

AR (I)	autoregressive source order one
BFOS	Breiman, Friedman, Olshen and Stone
bpp	Bits per Pixel
bpv	Bits per Vector
BQC	Block Quantisation Coding
BTC	Block Truncation Coding
CVQ	Classified Vector Quantisation
dB	decibel
DCT	Discrete Cosine Transform
DPCM	Differential Pulse Code Modulation
ECVQ	Entropy Constrained Vector Quantisation
FS-VQ	Finite State Vector Quantisation
FSVQ	Full Search Vector Quantisation
IAVQ	Image Adaptive Vector Quantisation
IC-IAVQ	Index Compressed Image Adaptive Vector Quantisation
IC-RAVQ	Index Compressed Residual Adaptive Vector Quantisation
IC-VQ	Index Compressed Vector Quantisation
IC-VQ (I)	Index Compressed Vector Quantisation method I
IC-VQ (II)	Index Compressed Vector Quantisation method II
IC-VQ (III)	Index Compressed Vector Quantisation method III
JPEG	Joint Photographic Expert Group
KLT	Karhunen Love Transform
MII	Map of Identical Indices

MSE	Mean Square Error
P_W	Probability of occurrence of blocks with identically indexed west-side neighbour
P_N	Probability of occurrence of blocks with identically indexed north-side neighbour
$P_{(W \cup N)}$	Probability of occurrence of blocks with identically indexed north- or west-side neighbour
$P_{(W \cap N)}$	Probability of occurrence of blocks with identically indexed north- and west-side neighbour
PECVQ	predictive entropy constrained vector quantisation
PPTSVQ	predictive pruned tree-structured vector quantisation
PSNR	Peak Signal to Noise Ratio
TC	Transform Coding
TSVQ	tree-structured Vector Quantisation
VB-VQ	variable block size Vector Quantisation
VQ	Vector Quantisation
WMSE	Weighted Mean Square Error

List of Figures

List of tables

Chapter 1

Introduction

1.1 Introduction

New communication and storage systems are geared towards utilising digital data because of its better amenability to regeneration, encryption, and less sensitivity to transmission noise than its analog counterpart. Furthermore, digital data also offers the possibilities of making better use of interference and noise-limited communication media [Clark 1985, Jayant 1984, Netravali 1981]. The rapid development in personal computers and digital processing capabilities in terms of size, speed, memory, price and the growth of computer based applications in areas such as medical imaging, archiving, surveillance, remote sensing and world wide databases are some other reasons to employ digital data.

The substitution of analog data with digital data requires larger bandwidth and more storage than analog signals [Clark 1985, page 2]. A frame of grey scale image, with the size of 512x512 pixels and considering 1 byte per pixel, requires 256 kbyte for storage. If one second of a movie consists of 24 frames, then one hour of the movie requires more than 22 giga bytes for storage or transmission. Large memory and channel capacity requirement make it mandatory to consider techniques that compress the data so as to use fewer bits for their representation. These schemes are called data compression, or source coding.

There are two types of image compression schemes; namely lossless and lossy. Those schemes, in which the original and the decompressed image are identical, are referred to as lossless, otherwise they are referred to as lossy. The quality of lossless coding schemes is measured by the ratio of the required bit rate for storing the original image to its compressed form (compression ratio), or the average number of bits required to represent each pixel of the image in compressed form (bit rate). The quality of lossy schemes is measured by the objective and/or subjective quality [Daumer 1982, Jayant 1984] at a given compression ratio or bit rate.

Low distortion in images may not be perceived by the human visual system, but provides means to improve the compression ratio in lossy schemes. These schemes can give significant compression when compared with lossless schemes. Therefore, in situations where some distortion is acceptable, the application of a lossy scheme is more appropriate. This is a motivation for the attention that has been devoted to the solution of the bandwidth conservation problem in communication and storage systems by lossy schemes.

There exists a variety of methods to achieve image compression in which some level of distortion is acceptable [Gersho 1992, Jain 1981, Jayant 1993, Kunt 1985, Netravali 1981, 1989]. The application of each scheme depends on the objective and subjective quality of the decompressed image, implementation and speed. Some applications require a simple and computationally efficient decoder, for example multi-receivers and archiving systems. In multi-receiver systems such as television broadcasting, it is most important that the receiver be simple so as to increase the number of viewers, thereby reducing the cost of the receiver. In archiving, the decoding

time is much more crucial than encoding, because decoding is usually required in real time, while encoding can be performed at off-time.

There are two groups of image coding schemes that have very simple decoders: schemes that operate on successive image pixels and those that operate on a block of image pixels. The schemes that operate on successive image pixels achieve compression by exploiting the inter-pixel correlation; an example is DPCM [Netravali 1977]. The second group of schemes are generally referred to as block coders in the literature, and exploit intra-block correlation to achieve compression. The second group possesses a better compression performance in comparison to the first group at the same level of distortion, because of their ability to exploit the correlation among a group of image block pixels rather than individual pixels [Clark 1985, Jain 1989, Gersho 1992].

One of the block coding schemes that has a very simple decoder is VQ [Gray 1984, Makhoul 1985]; the decoding process involves a simple lookup table. The encoder of a VQ maps a k-dimensional vector onto an integer (index) based on a fidelity criterion, and stores or transmits the index instead of the k-dimensional vector. The decoding procedure consists of finding the corresponding vector with the stored or received index from a table. While the attractiveness of VQ is its simplicity, the drawback of the basic VQ is its poor performance according to both objective and subjective fidelity criterion at low bit rates.

This thesis introduces two novel groups of techniques which significantly improve the objective and/or subjective quality of VQ at low bit rates. These schemes operate on the indices of vector quantised images, use

methods of non-stationary codebook or, variable bit assignment based on the activity of image blocks to achieve further improvement.

The first group of schemes compresses the indices to achieve performance improvement. The existence of high inter-pixel correlation in natural images leads to a high correlation among the indices of a vector quantised image. This correlation has been exploited by some lossless schemes. This thesis calls these schemes index compressed VQ (IC-VQ).

The second group of schemes employs a non-stationary codebook, or a variable rate coder in combination with IC-VQ schemes to improve the subjective quality of the vector quantised image at low bit rates. A problem of the basic VQ at low bit rates is the degradation of active areas such as edges; which results from the application of a stationary codebook for non-stationary sources or a uniform bit allocation disregarding the image area activity.

The most significant aspect of these new techniques is that they retain the benefits of the simplicity of traditional VQ and the speed of such fast VQ coders as the tree-structured VQ (TSVQ) [Buzo 1980]. There are other methods that exploit the inter-block correlation to improve the performance of VQ. However, the price is usually a design algorithm with increased complexity [Chou 1989, Foster 1985, Lookabaugh 1993, Nassrabadi 1990].

1.2 Overview of thesis

Besides the introduction, this thesis consists of five major parts spread over eight chapters. Chapter 2 is the background and literature review. Chapter 3 introduces the characteristics of indices. Chapters 4 to 6 present the new

methods of compressing the indices. Chapters 7 and 8 discuss existing methods of subjective quality improvement for VQ schemes at low bit rates, and present new methods in this regard. Chapter 9 contains conclusion and suggestion for further research. The subsequent paragraphs give a brief overview of each chapter.

Chapter 2 presents a literature review on VQ. The first section of the chapter looks at VQ as a family member of block coding schemes. The aim of this section is to introduce VQ in a wider context; compare it with other block coding schemes; present its advantages and disadvantages. This section discusses three block coding schemes, namely, transform [Clark 1985], block truncation [Delp 1979] and VQ. The second section of Chapter 2 discusses the methods employed to improve the performance of VQ at low bit rates. Two groups of these methods are some memoryless VQ techniques, such as classified and variable rate VQ, and VQ schemes with memory such as variable block size, predictive, adaptive, finite-state and address VQ. The approaches in each group are discussed. The final part of the second section presents the basic idea of index compression and its literature as a group of VQ schemes with memory.

The objective of Chapter 3 is to show the possibility of performance improvement of VQ schemes by using the characteristics of indices. Chapter 3 contains material which is the basis of the proposed coding schemes in this thesis, and the other chapters of the thesis extensively use the results obtained in this chapter. Firstly, Chapter 3 presents the analogy between the pixel and index domain correlation and, then shows the characteristics of indices empirically. A series of tests are performed on autoregressive sources and natural images to show the indices' characteristics.

Chapter 4 introduces a very simple and computationally efficient method of index compression based on the probability of identically indexed neighbouring blocks in a vector quantised image (index compressed VQ method I (IC-VQ (I))). Quantisation of highly correlated sources increases the probability of having neighbouring blocks with identical indices. If two neighbouring blocks have an identical index, there is no need to transmit the index for both of them. Two sets of information are sufficient to reconstruct all the indices: a map that shows the location of the identically indexed neighbouring blocks and the information about those blocks that do not have an identical index with their neighbours. Chapter 4 analytically shows the situation where IC-VQ (I) outperforms its counterpart VQ, and discusses the bit rates that IC-VQ (I) gives the maximum coding performance. The performance of IC-VQ (I) is compared with some VQ schemes and JPEG coding standard at low bit rates.

The coding performance of IC-VQ (I) depends on the probability of identically indexed neighbouring blocks. Chapter 3 shows that this probability decreases when the codebook size increases, hence an increase in the codebook size makes IC-VQ (I) less effective. Chapter 5 addresses the problem of IC-VQ (I) and introduces another method, index compressed VQ method II (IC-VQ (II)). The theoretical basis of this new technique is that the high inter-block correlation in natural images results in a high probability that neighbouring image blocks are mapped to codevectors which have high correlation. In other words, the neighbouring blocks are mapped to a very small subset of the codebook containing relatively highly correlated codevectors. It can be concluded that if instead of the whole codebook a small subset to encode the neighbouring blocks can be considered, it is

possible to improve the performance of traditional VQ schemes. Chapter 5 gives the rate formula for IC-VQ (II) and compares this scheme with IC-VQ (I), and some other coding schemes.

IC-VQ (II) allocates bits uniformly to image block indices by using a fixed size subset of the codebook to represent the neighbouring image block indices. This approach disregards the fact that high activity areas require a bigger subset of the codebook than low activity areas. In other words, high activity areas require more bits for their representation than low activity areas. Chapter 6 addresses the problem of uniform bit allocation employed in IC-VQ (II), and introduces index compressed VQ method III (IC-VQ (III)) to overcome this problem.

IC-VQ (III) compresses the indices generated from a TSVQ, and results in high performance improvement over the TSVQ scheme. TSVQ was first introduced [Buzo 1980] to alleviate the encoding complexity of the basic VQ, full search VQ (FSVQ), at the expense of more distortion than FSVQ. The combination of TSVQ with method III of index compression not only significantly improves the coding performance of TSVQ, but also this combination gives better performance than FSVQ.

The variable bit assignment procedure of IC-VQ (III) is based on considering the genealogy of indices. It is possible that neighbouring blocks might not have an identical index, but there exists the probability of having an identical parent, grandparents and so on. If two neighbouring blocks do not have an identical index for a given rate, a coarse quantisation may result in mapping them into the same index, or their indices belong to a

subtree of the big tree shaped codebook. IC-VQ (III) employs this feature to compress the indices.

Chapter 6 extends IC-VQ (III) for the case when the VQ coder is full search. The interesting characteristic of TSVQ is that each image block index contains its genealogical information, and acts as an identification card. This feature forms the basis of the method introduced in Chapter 6. The indices of a FSVQ do not carry the genealogical information, but through a simple method they can be easily endowed with such property, and consequently makes the combination of the third method of index compression with FSVQ possible. Chapter 6 performs some comparisons between IC-VQ (III) and some other coding schemes.

Chapters 7 and 8 focus on improving the subjective quality of VQ by exploiting the benefits of index compression at low bit rates. A method of improving the subjective quality of the coded image is to employ an image-based codebook rather than a universal-based codebook [Goldberg 1986]. The reason is that a stationary codebook cannot adequately represent non-stationary sources such as images. The method of image-based codebook VQ is sometimes referred to as image adaptive VQ (IAVQ). In Chapter 7 the application of index compression in combination with image-based codebook VQ schemes is investigated. The result of this synergy is compared with the basic IAVQ proposed by Goldberg et al [Goldberg 1986].

One of the problems of IAVQ is that it assigns bits to low and high activity areas uniformly. Such a uniform bit allocation results in assigning more bits than required to represent low activity areas which can be well represented by universal-based codebook VQ. Chapter 8 addresses the problem of bit

allocation in IAVQ, and introduces a method which first employs a universal based codebook VQ to encode all the image blocks and, then the residuals of active blocks are further encoded by an adaptive codebook. Chapter 8 introduces a simple method to detect the active blocks as well. This method employs the image block indices to distinguish the active blocks, and there is no need to transmit overhead information to represent the address location of the active blocks.

Chapter 9, the conclusion chapter, consists of two sections. The first section presents an overview of all chapters and the results obtained. The second section provides suggestions for furthering this study; additional research can have three directions, bit rate reduction of index compression schemes, application of index compression in areas such as medical images, and methods of bit allocation to improve the objective or subjective quality of the decompressed image.

1.3 Main Contributions

The main original contributions of this thesis are listed, the related papers to contributions are shown in front of them, and the complete information about papers is given in section 1.4.

1. The analogy between the index and pixel domain is presented in section 3.1 [8].

2. The possibility of reconstructing a low resolution version of an image using the image block indices is presented in section 3.2.2.1.

3. The high probability of having identically indexed neighbouring blocks, after vector quantising the image at low bit rates, is shown in section 3.2 [7,8].

4. A new coding technique, IC-VQ (I), which exploits the probability of identically indexed neighbouring blocks, is introduced in Chapter 4, and the condition under which IC-VQ (I) outperforms its corresponding VQ scheme is presented in section 4.4 [8].

5. The high probability that the vector quantised version of neighbouring image blocks are mapped onto a small subset of the VQ codebook is shown in section 5.2 [2].

6. A novel coding scheme, IC-VQ (II), which exploits the high probability of mapping the neighbouring blocks indices on a subset of the VQ codebook indices, is introduced in Chapter 5, and the optimum rate of IC-VQ (II) is given in section 5.4.1[2].

7. A novel variable rate VQ coder, IC-VQ (III), which exploits the genealogical relationship among the image block indices for the case of TSVQ is introduced in Chapter 6.

8. The concept of virtual tree for a FSVQ is introduced in section 6.7, and the new method developed for TSVQ in Chapter 6 is extended to FSVQ.

9. The application of index compression for a VQ scheme with memory to improve the performance of memoryless VQ is introduced (IC-IAVQ) in Chapter 7 [1, 5, 7, 9].

10. A simple and computationally efficient method of partitioning an image into high and low active blocks is introduced in section 8.3.

11. A method of variable bit allocation depending on image block activity in conjunction with index compression is introduced in Chapter 8 [2, 3, 6].

1.4 List of Publications

[1] J. Shanbehzadeh, P. O. Ogunbona, "Index Factorised Image Adaptive Vector Quantisation," International Symposium on Information Theory & its Applications, Sydney Australia, Vol. 1, pp. 187-190, Nov. 1994.

[2] J. Shanbehzadeh, P. O. Ogunbona, " New feature-based image adaptive vector quantization coder," International Conference on Coding and Signal Processing for Information Storage, SPIE's Photonics East Symposium, Philadelphia, PA, Proceeding 2605, Paper No. 19, 22-26 Oct. 1995.

[3] J. Shanbehzadeh, P. O. Ogunbona, " Probability Constrained Vector Quantisation for Low Bit Rate Image Coding," Proceedings Digital Image Computing Techniques and Applications," Brisbane, pp. 127-132, Dec. 1995.

[4] J. Shanbehzadeh, P. O. Ogunbona, " Image Adaptive Vector Quantisation Based on Visual Perception," Proceedings Digital Image Computing Techniques and Applications," Brisbane, pp. 127-132, Dec. 1995.

[5] J. Shanbehzadeh, P. O. Ogunbona, "A Comparative Study of Image Adaptive Vector Quantisation Schemes," The Proceeding of the 6th International Conference on Signal Processing & Technology, Vol 1, Boston, Ma, USA, pp. 945-948, Oct. 1995

[6] J. Shanbehzadeh, P. O. Ogunbona, "Visual perception based image adaptive vector quantisation for low bit rate applications," submitted to the Journal of Electronic Imaging, Nov. 1995

[7] J. Shanbehzadeh, P. O. Ogunbona, "Index compressed image adaptive vector quantisation," Signal Processing: Image Communication, to appear in 1996.

[8] J. Shanbehzadeh, P. O. Ogunbona, "Index compressed vector quantisation," submitted to IEEE Trans. on Image Processing.

[9] J. Shanbehzadeh, P. O. Ogunbona, "On the Comparative Study of the Codebook Generation Time of LBG and PNN Algorithms," submitted to IEEE Trans. on Image Processing.

Chapter 2

Literature Review

2.1 Introduction

This chapter consists of two sections. The first section is a brief discussion about a group of image compression schemes known as block coding. Three members of this group- orthogonal transform coding (OTC), block truncation coding (BTC) [Delp 1979] and VQ- are presented, and their differences, advantages and disadvantages are discussed. The second section is a literature survey on the methods proposed for improving the performance of VQ at low bit rates, and expounds the idea and the literature of index compression, which is the centre of attention in this thesis.

2.2 Block Coding

Block coding generally refers to those image compression schemes that compress a block of image pixels at each step. These schemes can be divided into two sets (Figure 2.1), namely orthogonal transform coding (OTC) and block quantisation coding (BQC). The compression capability of these schemes derives from the exploitation of the intra-block correlation. OTC

employs an orthogonal transform [Ahmad 1975], and BQC quantises an image block in the pixel domain to exploit the inter-block correlation.

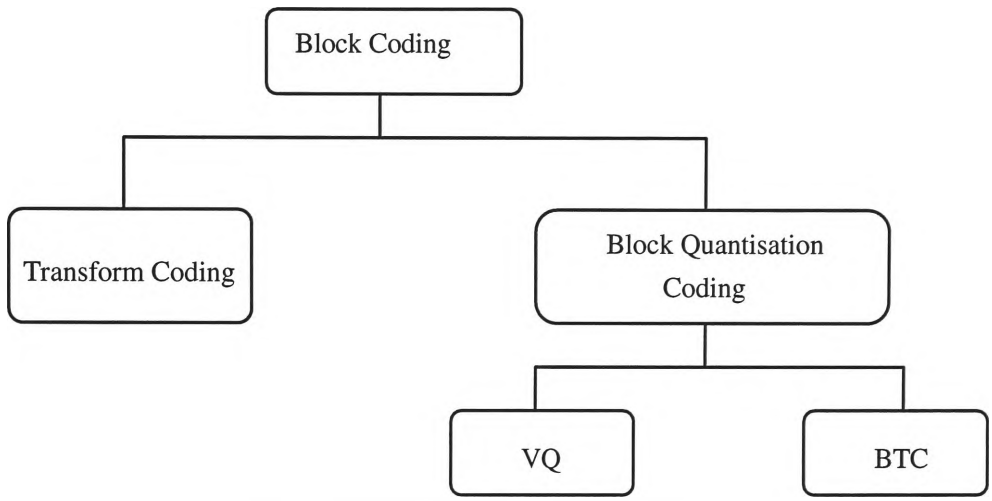

Figure 2.1: Block coding schemes

BQC schemes can be further divided into two groups, BTC and VQ. The difference between these two methods is in their quantisation schemes. BTC employs a scalar type quantisation based on the block pixels characteristics, and VQ quantises the block of pixels as one element.

2.2.1 Orthogonal Transform Coding

OTC is based on applying an orthogonal transform [Ahmed 1975] to each of the image blocks. The outcome of such transformations for images is a packing of most of the image block's energy in just a few transform domain coefficients (energy packing property). Hence, it is possible to approximately reconstruct an image block by just transmitting or storing the few transform domain coefficients that contain most of its energy.

There are several orthogonal transforms, namely the discrete Karhunen-Loeve, Fourier, Cosine, Sine, Slant, Walsh, Hadamard, Walsh-Hadamard and Haar transforms. The optimum orthogonal transform in terms of energy packing efficiency is the Karhunen-Loeve transform (KLT) [Ahmed 1975],

but the complexity involved in computing the KLT matrix makes it impractical [Rao 1990 pp. 31]. Besides, the energy packing superiority of KLT for highly correlated sources over some other transforms is marginal [Clark 1985 pp. 95]. From a practical point of view, the best transform in terms of energy packing capability for highly correlated sources is the discrete cosine transform (DCT) [Clark 1985 page 130, Rao 1990 page 123]. This conclusion has resulted in the development of fast algorithms and hardware implementations for the DCT [Rao 1990 pp. 48, pp. 439-446].

Figure 2.2 shows the block diagram of a simple OTC scheme. In the encoder section, the image is first partitioned into square blocks, then each block undergoes a linear orthogonal transform, and the resulting transform domain coefficients are allocated bits optimally [Huang 1966] and scalar quantised. The quantised coefficients are transmitted or stored. In the decoder section, the coefficients are first dequantised, and then the inverse transformation produces an approximate version of each image block. The most important parts of an OTC, which results in the most compression, are the bit allocation and quantisation step [Clark 1985]. The bit allocation process is based on the importance of the transform domain coefficients, and the quantisation puts the coefficients in a suitable form for transmission.

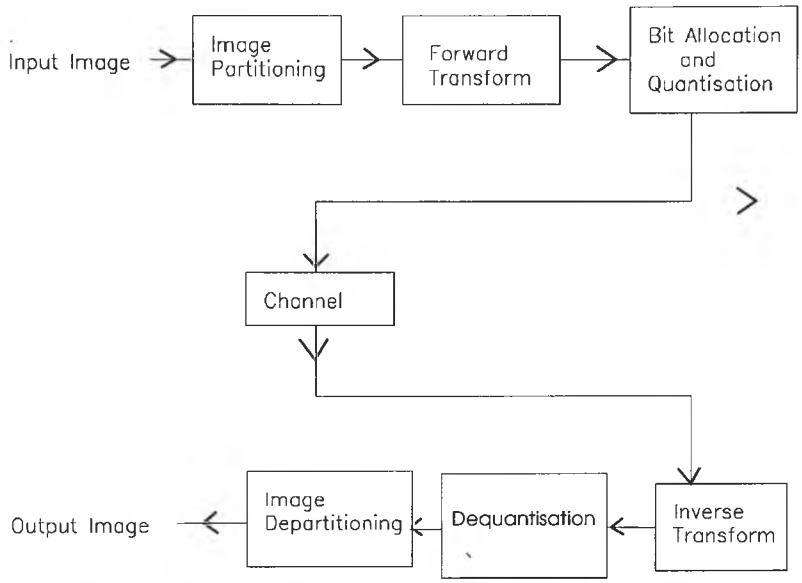

Figure 2.2: Block diagram of a compression scheme based on transform coding

OTC suffers from blocking effects at low bit rates. The reason is the independent processing of each image block. Some of the coding errors generated from quantisation will produce discontinues in the image in a way that the border pixels of one image block will, most likely, not match with the border pixels of the next image block, thus generate the blocking effect [Rao 1990, Netravali 1989]. Psychological considerations show that the block-shaped distortion generated from orthogonal transforms is ten times more displeasing than random noise distortion [Miyahara 1985].

2.2.1.1 Joint Photographic Expert Group Coding Standard

After about 30 years of research on OTC, a modification of this scheme has become a standard still image compression technique under the name Joint Photographic Expert Group (JPEG) [Wallace 1991]. Here, the JPEG standard is briefly described for two reasons- firstly its high performance over the traditional transform coding schemes, and secondly most researchers in image coding use the JPEG standard as a benchmark to evaluate their results [Weiping 1995, Egger 1995, Constantinescu 1994, Franti 1994]. JPEG has two advantages over traditional OTC schemes:

adaptive quantisation, and a variable bit assignment procedure which matches the characteristics of each image block activity. Figure 2.3 shows the block diagram of a JPEG encoder.

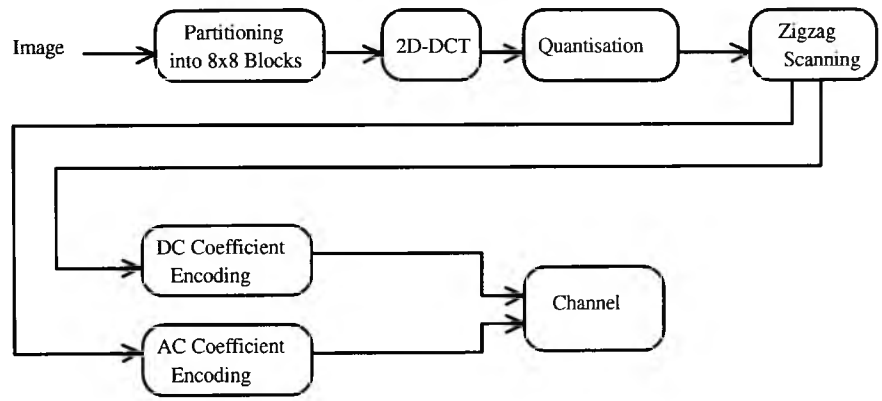

Figure 2.3: Block diagram of a JPEG encoder

The JPEG encoder first partitions the input image into 8x8 blocks of pixels and then applies a two dimensional DCT to each image block. The transform domain coefficients are scalar quantised in a simple procedure of dividing each coefficient by a scalar, and rounding the resulting number. Despite the simplicity of this process the quantisation is adaptive in the sense that each coefficient is quantised based on its effect on the human visual system. The coefficients of each image block are scanned in a zigzag manner to increase the possibility of runs of zeros. The DC coefficients, which are the first coefficient of each block, and AC coefficients are separately encoded. The encoding process of DC coefficients is a form of differential coding of the row scanned version of the DC coefficient of blocks. The AC coefficients undergo an adaptive Huffman [Huffman 1952] (or arithmetic [Pasco 1976, Rissanen 1981, Jones 1981]) coding and are transmitted. (In a baseline JPEG coder Huffman coding is mandatory.) The decoder section is simply the inverse of the encoding process. A more detailed description of the JPEG standard can be found in [Wallace 1991, Pennebaker 1993].

The quality of the coded image by JPEG at a given rate is highly dependent on the image activity and the bit rate. JPEG assigns more bits to active areas presumed to be important to the human viewer. However, because of the universal applicability of the JPEG coding standard, in some cases this method of quantisation results in assigning more bits to active areas such as textures like hair, grasses and fur, which are not important to the human viewer. On the other hand less bits are assigned to low activity areas. This approach results in blocking effects especially in low activity areas of the image at low bit rates.

2.2.2 Block Truncation Coding

The basic block truncation encoder first partitions the image into non-overlapping similarly sized blocks, and extracts the mean and standard deviation of each block. The encoder constructs a bit plane for each block by setting the image block pixels with less than the block mean to zero and the rest to one. The basic block truncation decoder requires information of the mean, standard deviation and the bit plane of each image block for reconstructing the image. The image blocks are reconstructed by substituting the zeros and ones of the blocks' bit plane with a and b that are calculated according to the following formulae:

$$a = \bar{x} + \sigma\sqrt{\frac{q}{m-q}} \tag{2.3}$$

$$b = \bar{x} + \sigma\sqrt{\frac{m-q}{q}} \tag{2.4}$$

where $\bar{x}$, σ, m and q are the block mean, standard deviation, number of block pixels, and the number of block pixels assigned a one respectively. If k bits are assigned to each of the two factors of a block, standard deviation and mean, then the required rate for BTC is:

$$R_{BTC} = 1 + \frac{2k}{m} \text{ bpp} \tag{2.5}$$

If 8 bits are assigned to mean and standard deviation, the rate will be 2 bpp.

BTC is a very simple and computationally efficient coder, because the encoding process consists of calculating the mean, standard deviation, two factors (a and b), and a bit plane, and the decoding process consists of substituting the bit plane elements with two values (a and b). However, the basic BTC has a low compression ratio, and cannot reproduce active areas such as edges well [Franti 1994]. In the literature, efforts to improve the performance of BTC are in the direction of reducing the required rate to represent the mean and standard deviation or the bit plane [Franti, 1994, Franti 1995, Healy 1981, Wu 1991, Udpikar 1987, Lu 1991, Mitchell 1980, Nasiopoulous 1991, Kamel 1991, Roy 1991]. In literature, there is no application of BTC at bit rates less than 1 bpp, and its successful application at bit rates around 1 bpp requires using more complex procedure which detract from the significant feature of the basic BTC; its simplicity [Franti, 1994].

2.2.3 Vector Quantisation

The basic VQ [Gray 1984, Makhoul 1985, Nasrabadi 1988] encoder first partitions the input image into small and similarly sized blocks (vectors), then compares each vector with a set of pre-determined vectors (codebook), and finds the best matching with the vector from the codebook in terms of a similarity measure (mostly MSE, or WMSE [Cosman 1993]). Associated with each vector in the codebook (codevector) is an index, and the encoder stores or transmits the index of the best matching codevector instead of the image block. The basic VQ decoder reconstructs the vectors by substituting the image vectors' indices with their corresponding codevectors from the

codebook. VQ achieves its compression capability by representing a block of pixels with the corresponding index in the codebook. The bit rate of the basic VQ is the ratio of the bits to represent the index of a vector over the vector dimension [Gray 1984, Makhoul 1985, Nasrabadi 1988].

The most interesting feature of the basic VQ coder is its decoder simplicity. The decoder part is just a lookup table. Each index indicates the address of its corresponding codevector in the codebook. It can be seen that VQ has a simpler decoder than OTC, and BTC. The encoding process of VQ involves more operations than BTC, because the VQ encoder has to compute the MSE between each block and all the members of a set of codevectors, and then selects the codevector with the smallest MSE. The BTC encoder only needs to compute the variance and mean of a block, and based on these two factors quantises the block as explained previously.

The implementation of a VQ scheme involves a distortion calculation for each image block and comparisons to select the least distortion- whereas a transform coder requires a transformation and a quantisation. Whichever implementation is selected- software or hardware- distortion calculations and comparisons are much simpler computational operations than an orthogonal transformation. Of course one needs all the attending factors to have an accurate comparison of the computational time.

The encoding time of VQ depends on the rate, while the encoding time of OTC and BTC schemes have no dependence on rate and always remain the same. At low bit rates the computational cost of VQ is considerably reduced, because the codebook size is small, thus the number of distortion

calculations is small. By comparison there is no computational change for OTC and BTC.

2.3 Review of the Methods of Improving the Performance of the Basic VQ at Low Bit Rates

The basic VQ has to represent all the possible forms of image blocks by only a small set of fixed vectors at low bit rates, thus its performance is poor [Nassrabadi 1988]. Two important methods of improving the performance of VQ at low bit rates are some memoryless VQ schemes such as classified VQ (CVQ) [Ramamurthi 1986] and variable rate VQ [Chou 1989, Lookabaugh 1993, Riskin 1990, 1991], and VQ schemes with memory such as variable block size VQ (VB-VQ) [Vaisey 1988, 1992], finite state VQ (FS-VQ) [Aravind 1986, 1987], predictive VQ [Lookabaugh 1993], image adaptive VQ [Goldberg 1986], and index compressed VQ (address VQ [Feng 1988, Nasrabadi 1990a, Nasrabadi 1990b]). This section presents each of these methods briefly.

2.3.1 Memoryless VQ schemes

Memoryless VQ schemes refer to those techniques which disregard the inter-block dependency in the compression process. The notable methods are CVQ and variable rate VQ. Next each of these schemes are explained.

2.3.1.1 Classified VQ

One of the disadvantages of the basic VQ schemes, such as full search and tree-structured VQ, is their inability to reproduce edges at low bit rates [Nasrabadi 1988]. Ramamurthi and Gersho [Ramamurthi 1986] introduced classified VQ (CVQ) to alleviate this problem. CVQ partitions the image

blocks into edges with various orientations and background classes, and generates separate codebooks for each class. This scheme assigns more bits and uses a smarter codebook to compress the edge information.

The advantage of CVQ, besides its good ability in edge reproduction, is its reduced computational cost when compared with FSVQ. In order to improve the performance of VQ, it is necessary to increase the codebook size and thus the computational cost of encoding. CVQ obviates this problem by requiring a small codebook for each class of image block. However the drawback of CVQ is the overhead needed to identify the class of each image block. Figure 2.4 shows the block diagram of this scheme.

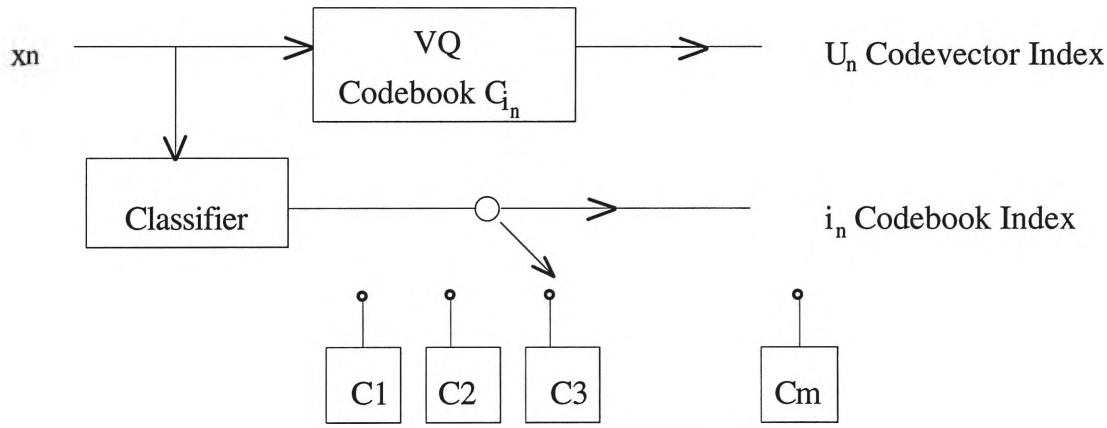

Figure 2.4: Block diagram of CVQ

CVQ results in images with good quality at bit rates above 0.7 bpp [Ramamurthi 1986]. Some modifications of CVQ with smarter block classifiers in the transform domain have been reported which are able to encode images with good quality at bit rates down to 0.4 bpp [Kim 1991, Kim 1992]. The improvement has been achieved at the expense of applying a linear orthogonal transformation in the encoder which consequently increases the computational burden of the encoding process.

2.3.1.2 Variable rate VQ

One of the problems of the basic VQ is the assignment of the same amount of bits to all the generated clusters, whereas the clusters normally have different probability of occurrence and different effect on the incurred distortion. A straight forward method of improving VQ performance is to use a variable rate based on the clusters' probability of occurrence and their effects on the distortion.

Two groups of methods have been introduced in this regard; entropy constrained VQ (ECVQ) and pruned TSVQ [Chou 1989, Lookabaugh 1993, Riskin 1990, Riskin 1991]. In ECVQ [Chou 1989], a variable length code is assigned to each codevector depending on the distortion and the probability of occurrence of a cluster member. In pruned TSVQ, a balanced tree [Lookabaugh 1993, Riskin 1990] or a greedy tree [Riskin 1991] is optimally pruned by Breiman, Friedman, Olshen and Stone (BFOS) algorithm [Breiman 1984].

The improvement achieved by pruned TSVQ (PTSVQ) and ECVQ over FSVQ at 0.3125 bpp for a test image from USC database "*Lena*" are only 0.23 dB and 0.5 dB and in maximum cases are 0.5 dB and 1.7 dB respectively. In terms of subjective quality both schemes have the same problems as the basic VQ at low bit rates. The only advantage of PTSVQ over FSVQ is its tree-shaped searching process which makes PTSVQ computationally more efficient than FSVQ. The design process of ECVQ is much more complex than FSVQ [Lookabaugh 1993 page 191, Gersho 1992 page 665].

In general variable rate coding improves the performance of a fixed rate coder [Gersho 1992, pages 631-632], and they are more suitable for applications such as storage and packet switched communication networks. However,

there are two problems in application of variable rate coders. First, many applications require a fixed transmission rate. In these situations, extra buffering equipment is required to change a variable rate coder to a fixed rate coder. Secondly, some variable rate coders propagate the channel noise. For example, in variable length coding techniques, a change in any of symbols may result in considering a longer or shorter code instead of the real code, and this error propagates through the entire bit stream.

2.3.3 VQ schemes with Memory

Memoryless VQ coders are sub-optimum in the sense that they disregard the inter-block correlation. In natural images there exist high inter-block correlation that can be exploited to achieve further coding performance. The group of VQ schemes in which inter-block correlation is exploited are called VQ with memory.

Some important groups of VQ schemes with memory which have been used in image compression are variable block size, predictive, finite state, image adaptive and address VQ. Here, each of these schemes is briefly discussed. While all these schemes outperform FSVQ at low bit rates, they introduce more complexity in design or implementation.

2.3.3.1 Variable block size coding

In a block coder such as VQ, the first step is to partition the input image into small blocks, where in most cases the blocks are distinct and are the same size. There are big smooth areas in images that can be compressed with a small fraction of bits, and the small block size partitioning ignores this characteristic. A method of exploiting this characteristic is to partition the

image into variable size blocks, where the bigger blocks are in smooth areas and the smaller ones are in active areas.

Vaisey and Gersho [Vaisey 1988, 1992] introduced a variable block size coder based on VQ which results in a very good subjective quality at 0.325 bpp. However their scheme requires three more steps besides a variable block size partitioning process. A hybrid transform VQ is used to encode large-sized blocks, and a transform-domain-based method is employed to detect the high activity blocks, which are of low visual importance. Finally a post-processor is utilised to improve the subjective quality of the quantised image. These several steps make their algorithm tremendously complicated.

One of the disadvantages of variable block size VQ is the requirement of a scheme to divide the image into variable size areas in a way such that highly correlated regions have a large size and low correlated regions have a small size. This scheme requires a criterion to measure the correlation in each part of the image or a simple method such as AC energy of blocks to meet that requirement. The later results in an increased computational complexity of the scheme. The former results in an approximate scheme which may not been able to distinguish the visually important areas from the areas of low importance, since the visually important areas may also be located in low activity regions. Another disadvantage of variable block size VQ is the overhead information about the size and location of blocks. Although there exist fast and efficient methods in this regard [Samet 1984], there is still the constraint imposed by the allowable shape of blocks which leads to inefficient use of bits [Boxerman 1990].

2.3.3.2 Predictive VQ

Prediction is the statistical procedure of estimating one or more random variables from the observation of other random variables. In highly correlated images the predicability of past or future image pixels or blocks from the present values is considerable. No cost in terms of bits needs to be paid for prediction, and the residual errors between the predicted and actual pixels values require less bits than the original values for transmission or storage . Herein lies the main advantage of predictive coding. So far there is no theoretical proof to show that predictive VQ performs better than memoryless VQ, but this is the case in practice [Gersho 1992].

Predictive VQ first estimates an image block from its neighbours, then finds the residual error between the estimated block and the actual one. The index of the residual vector is transmitted or stored. For simplicity, three neighbouring blocks are normally used to predict an image block. An example is shown in Figure 2.5, pixel "*x*" of the image block, which is under estimation, is predicted by using five pixels from its neighbouring blocks, namely pixels *a, b, c, d,* and *e*. Figure 2.6 shows a block diagram of a predictive vector quantiser.

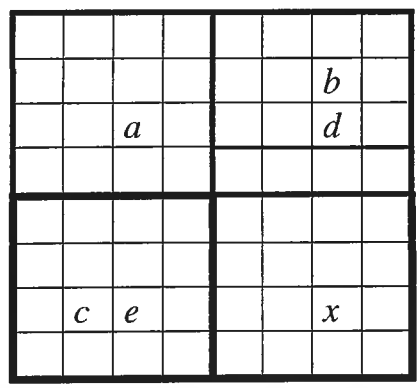

Figure 2.5: Predicting pixels of the block in south-east from its neighbouring blocks

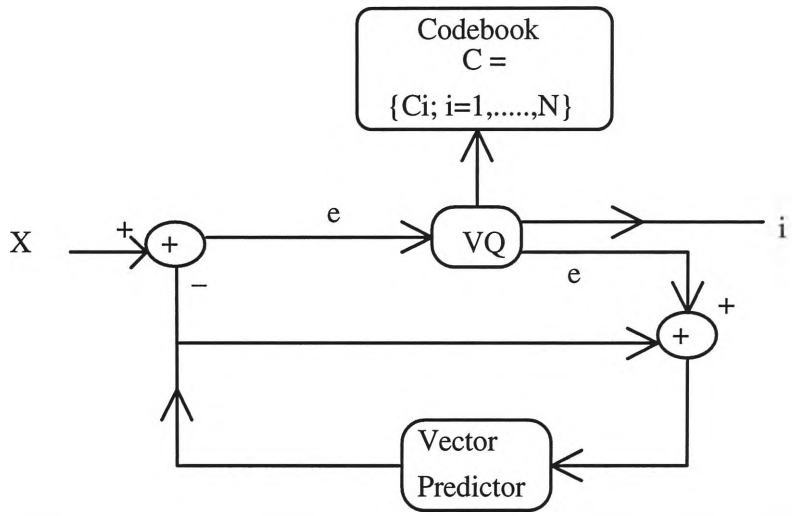

Figure 2.6.a: Encoder of a Predictive Vector Quantiser

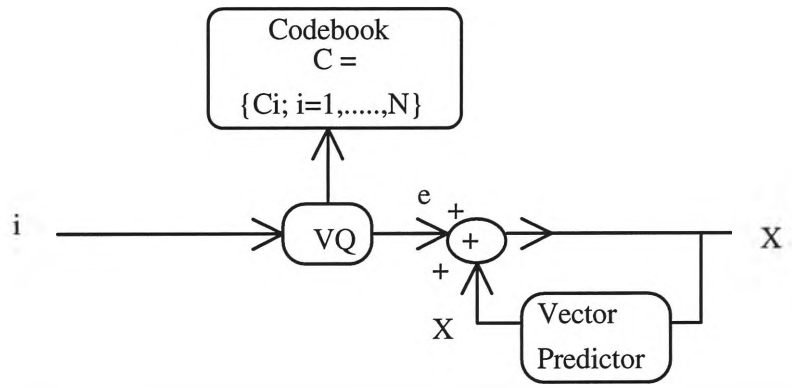

Figure 2.6.b: Decoder of a Predictive Vector Quantiser

A group of VQ coders are variable rate predictive coders such as predictive entropy constraint VQ (PECVQ), and predictive PTSVQ (PPTSVQ). These schemes have both the advantages of variable rate and predictive coders. The performance of these predictive coding schemes show significant improvement over FSVQ both objectively and subjectively for a test image "*Lena*" at bit rates around 0.3 bpp. Of course these advantages are achieved at the use of more complexity in design and implementation when compared to the basic VQ [Lookabaugh 1993 page 193].

2.3.3.3 Finite State VQ

Finite state VQ (FS-VQ) is another member of the VQ family of schemes with memory. From one point of view FS-VQ is similar to CVQ. The encoder of CVQ finds the class of each block and, the best match with the

block from its class. It thus requires two groups of information, one to identify the class of the block and the other for the best match with the block within its class. The procedure for FS-VQ is different: instead of a class, each block has a state, and each state has a codebook. Encoder and decoder predict the state of each block by considering its previous blocks, thus eliminating the need for extra information about the state of each block. The FS-VQ encoder finds the best match with each block within the members of the block state, and the FS-VQ decoder finds the associated codevector to each image block index by searching through the corresponding state codebook. It is important that the decoder and encoder should always be synchronised, because there is no overhead to introduce the next state, and the decoder has to find it itself (next-state function block in Figure 2.7).

Figure 2.7 shows the block diagram of the encoder and decoder of a FS-VQ. In the figure S_n is the state of a block, C_i represents the codebook of each state, and U_n is the index associated with a state. The encoder finds an image block index based on the block's state. A next-state function uses the index of the encoded vector to generate the state of the next vector. The decoder uses the index of a previously received vector to generate the state of the newly arrived vector. The state of a vector and its index are used to find its corresponding codevector.

The design process of FS-VQ is complicated. It involves designing an initial classifier to classify the input vectors into states, a state space, next state function and the state codebook. The state codebook should be improved by some iteration process. It is obvious that the design procedure of FS-VQ is more complicated than the basic VQ which requires only a codebook generation algorithm and a set of training vectors.

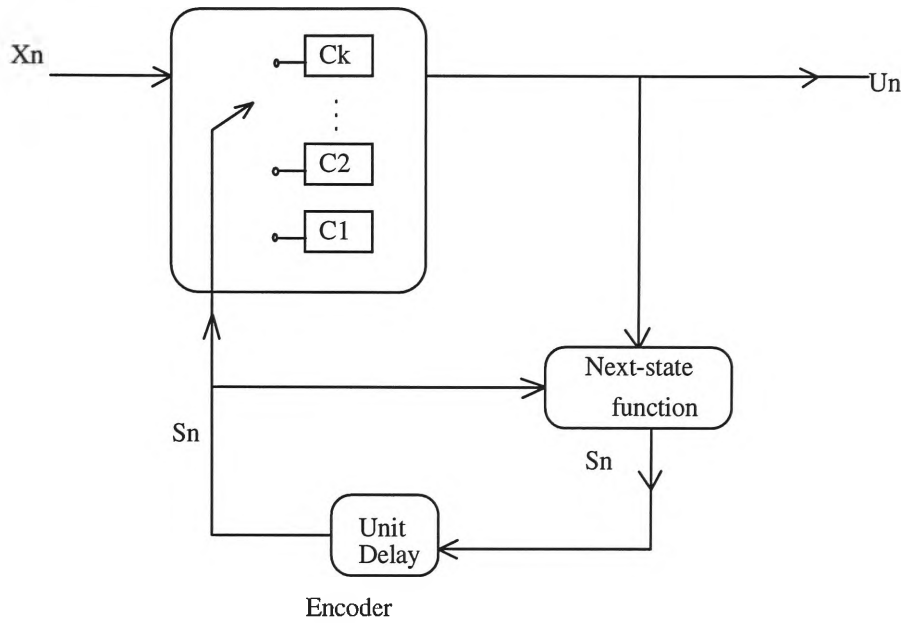

Figure 2.7.a: Encoder of a FS-VQ

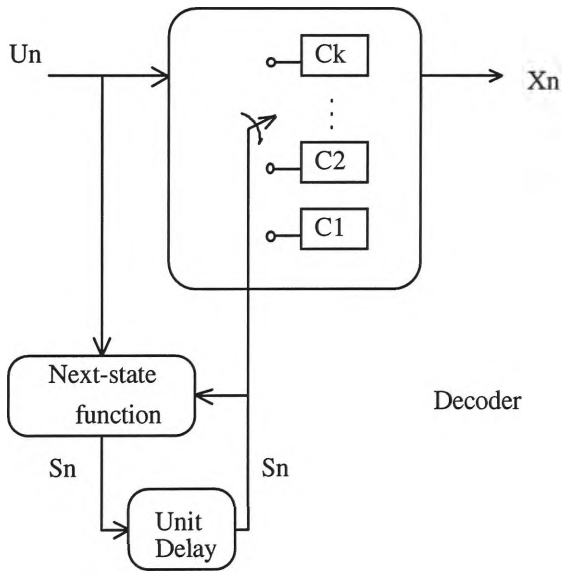

Figure 2.7.b: Decoder of a FS-VQ

The application of FS-VQ in image compression was first introduced by Aravind and Gersho [Aravind 1986, 1987]. Excellent results have been achieved at 0.25 bpp for the "*Lena*" image (30 dB PSNR) when FS-VQ in combination with Huffman coding is used [Kim 1988]. A drawback of FS-VQ, apart from its design complexity, is its high sensitivity to noise. If the encoder and decoder lose their synchronisation, the rest of the image blocks from that point onwards is corrupted by noise.

2.3.3.4 Image Adaptive VQ

One of the disadvantages of the basic VQ is the use of a stationary codebook for non-stationary sources such as images. This results in poor representation of active areas such as edges [Goldberg 1986]. Better performance in compression can be achieved if one uses the characteristics of each image rather than a general assumption [Gersho 1992, Jayant 1993].

A method of exploiting the image specification in a compression procedure is to generate the codebook based on the image itself [Goldberg 1986] or to update the codebook as the characteristics of the images under consideration changes [Gersho 1989]. The side effects of adaptive codebook generation and codevector updating are the overhead information about the codebook that needs to be transmitted, and real-time codebook generation.

2.3.3.5 Address VQ and Index Compressed VQ

So far two groups of methods to improve the performance of VQ have been presented. The first group, memoryless VQ schemes, improves the coding performance of VQ by allocating more bits to active or visually important areas such as edges, or by assigning more bits to areas with low probability of occurrence and less bits to others. The second group, VQ schemes with memory, is based on exploiting the inter-block correlation, non-stationary codebook or using a variable rate coder in combination with inter-block correlation removal schemes.

Another approach is that which exploits the inter-block correlation in the index domain. This scheme is a combination of a VQ scheme and an a lossless index compression scheme. Feng and Nasrabadi [Feng 1988,

Nasrabadi 1990a, Nasrabadi 1990b] proposed this scheme for the first time and they called it address VQ. In the form in which it was proposed, address VQ enjoyed the simplicity of memoryless VQ but suffers from a complexity problem in the compression of the indices and codebook generation for encoding the indices [Gersho 1992]. Albeit, very good results are reported at low bit rates.

The index compression scheme introduced by Feng and Nasrabadi [Feng 1988] has three codebooks. The first codebook is used to quantise a mean removed version of the image blocks. The size of this codebook is 128. The indices obtained are then sent into a lossless vector quantisation scheme based on the second codebook. The indices of each four neighbouring blocks are compared to a very big codebook of patterns; the size of the used pattern is 16384. If there exists a match with the indices of these four neighbouring blocks they will become candidate for another test, otherwise the index of these blocks has to be transmitted. If there exists a match, instead of 4 indices, one index from the second codebook can be transmitted.

A third step is performed on the indices of those four neighbouring blocks which has a matching index from the second codebook. In this step the indices of 4 nearest neighbouring blocks, 16 neighbouring blocks in the form of 4x4, are compared with the patterns in the third codebook. If there is a match, then instead of the indices of 16 blocks, one index is sufficient. Nasrabadi and Feng used a codebook of size 1024 for the third step. A map is required to show the results of the test performed in each section.

The first problem of address VQ is the high computation required to generate the second and third codebooks. For a perfect match in the second

codebook, when the size of the first codebook is 128, its size should be 268435456, and consequently the size of the third one is out of the question. However, because of high correlation in natural images, the probability of having most of the combinations of 4 out of 128 indices associated with the codevectors is low, and there is a need just to select some of the combinations with high probability of occurrence and ignore the rest. This requires reordering all the occurred combinations based on their frequency. However, this procedure demands a high computational cost because of the large number of the possible combinations. In a very similar method, Nassrabadi and Feng [Nasrabadi 1990b] solved the computational complexity, but their solution provided other problems such as the synchronisation between the encoder and decoder, and the computational complexity of reordering the second codebook at the receiver and transmitter during the encoding of each block [Nasrabadi 1990b].

The second problem of address VQ as proposed by Feng and Nasrabadi [Feng 1988, Nasrabadi 1990a, Nasrabadi 1990b] is the method of predicting the mean of blocks. They used the dequantised version of previous blocks to predict the mean of blocks without transmitting any information. In other words, the mean of blocks highly depends on the mean of previous blocks. The side effect of this method is a reduction in the dynamic range of the block means, and a consequent blurred image.

The third problem of address VQ is the amount of memory required for storing codebooks in comparison to its counterpart mean removed VQ. Assuming 8 bpp for each component of the codevectors of the basic mean removed VQ, the ratio of the codebooks for the address VQ to the basic

mean removed VQ is about 36. If image quality is to be improved by increasing the codebook size, this ratio will increase considerably.

In spite of all the enumerated problems, the high performance of address VQ in terms of PSNR at low bit rates, makes it feasible to apply VQ at low bit rates. The PSNR of address VQ for the same image used in other previous sections, *Lena*, is 30.6 dB at 0.256. This thesis introduces new VQ schemes based on index compression- they are termed index compressed VQ (IC-VQ) scheme. All these schemes compress the indices obtained by a FSVQ or TSVQ and use only the codebook of a simple VQ scheme such as TSVQ and FSVQ.

The basis of the compression capability of the new method is the high inter-block correlation in natural images. This affects the characteristics of the indices of neighbouring blocks such that the neighbouring blocks tend to have identical indices or are mapped onto a small subset of the set of all the VQ indices. The difference between the newly introduced index compression schemes and address VQ is the direct use of image indices' characteristics to exploit the inter-block correlation, rather than generating a global set of vectors to describe the inter-block correlation as in address VQ. In other words the index compression methods proposed in this thesis are image adaptive.

Figure 2.8 shows the generic block diagram of the new coder. At first the input image undergoes a VQ scheme, then the resulting indices are compressed and sent through the channel. At the decoder, the indices are decompressed and the image is reconstructed by a simple look-up table.

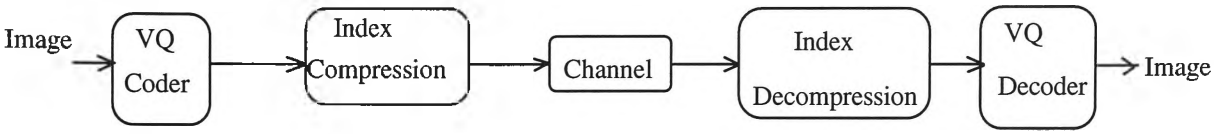

Figure 2.8: Block diagram of an IC-VQ scheme

Despite the overall simplicity of the scheme, its performance, especially at low bit rates, is very interesting and better than most of the VQ coders with memory, and it is objectively and subjectively superior to the JPEG coding standard. Some of the new index compression schemes, with little modification, give about the same results as address VQ, but with much less complexity in terms of codebook generation, encoding, decoding and the required memory for the codebook than address VQ.

2.4 Summary

This chapter first presented some block coding schemes including orthogonal transform coding, block truncation coding and VQ. These schemes are compared in terms of implementation and coding performance. It is shown that the decoder part of VQ is computationally more efficient than the other block coding schemes and that VQ shows better performance than BTC at low bit rates.

The second section reviewed the methods for improving the performance of VQ at low bit rates. Some memoryless VQ coders, such as CVQ, and VR-VQ, by generating special codebooks for visually important areas or assigning more bits to active areas, improve the coding performance of the basic VQ marginally. VQ schemes with memory are able to improve the performance of the basic VQ coders by considering the inter-block correlation, but at the expense of more complexity in the design procedure.

The final section focused on a group of VQ schemes with memory, called index compression schemes, which exploit index correlation rather than directly operating on the image blocks pixels as other VQ schemes with memory. An early approach to index compression, address VQ, provides significant results but is computationally complex and expensive due to codebook generation and codebook memory requirements respectively. This thesis introduces some new index compression schemes. The new schemes are able to provide comparable results with much lower complexity than address VQ.

Chapter 3

Characteristics of Quantised Image Indices at Low Bit Rates

3.1 Introduction

This chapter presents the characteristics of the indices obtained from a quantised image at low bit rates. It is shown that the high correlation among image pixels exhibits itself among the indices of the quantised image. An outcome of this feature is the high probability of identically indexed neighbouring pixels or vectors (blocks). The new compression schemes introduced in this thesis extensively employ this feature to compress the indices.

The organisation of this chapter is as follows. The next section establishes the connection between correlation in pixel and index domains. The second section empirically shows the probability of having two neighbouring pixels or vectors with identical indices is high. The empirical results have been obtained from tests carried out on a source model and some USC database images. This chapter separates the treatment of scalar from the vector case for ease of presentation, since the latter is a generalisation of the former.

3.2 Image Correlation in Pixel and Index Domains

The scalar or vector quantisation encoder compares a pixel or a vector to the members of a codebook, and selects the best matching pixel or vector from the codebook (codevector); the encoder transmits or stores the associated index. This section shows that the quantisation of highly correlated sources results in highly correlated indices. The following two sub-sections demonstrate this characteristic for scalar and vector quantisation.

3.2.1 Pixel and index domains correlation in scalar quantisation

The effect of scalar quantisation on the index domain correlation can be easily investigated by considering an N-point memoryless uniform scalar quantiser. While a uniform scalar quantiser is not the optimum scalar quantiser, it satisfies the present purpose to demonstrate the relationship between pixel and index domains correlation.

An N-point uniform scalar quantiser partitions a continuous subset of real numbers into N exclusive cells, assigning a real and an integer number to each cell; the real number is the quantised version of each member of the cell and the integer number is its associated index. Either of these two values is sufficient to specify the quantised value of each real number.

A test has been performed on an autoregressive source order one, *AR(1)*, to find out the relationship between the correlation in pixel and index domains. An *AR(1)* model is useful to represent a raster scanned image [Jain 1989, page 190-194], and is given as:

$$x(n) = rx(n-1) + \varepsilon(n) \tag{3.1}$$

where $x(n)$, r, and $\varepsilon(n)$ are the signal, the first lag correlation coefficient and the white noise signal respectively. N-point uniform scalar quantisers, depending on r and for N between 1 and 256, have been generated. Figure 3.1 shows the results of the indices correlation for highly correlated sources. Figures 3.2 to 3.4 depict an *AR(1)* source with r=0.96, its quantised version at 4 bpp and their associated indices respectively.

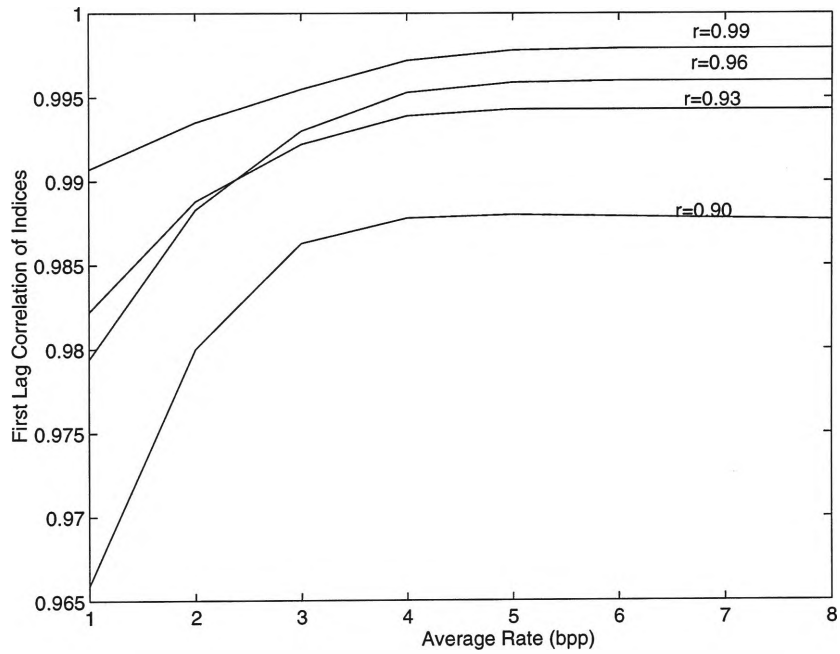

Figure 3.1: Effect of uniform scalar quantisation on the first lag correlation coefficients of the indices of an *AR(1)* source

Figure 3.1 shows that the quantisation of highly correlated sources generates highly correlated indices. The reason can be stated as follows. Highly correlated sources contain pixels with about the same value. This characteristic increases the probability of mapping the scalar quantised version of two neighbouring pixels into the same quantisation cell, or into the neighbouring cells which are relatively highly correlated. This leads to high indices correlation.

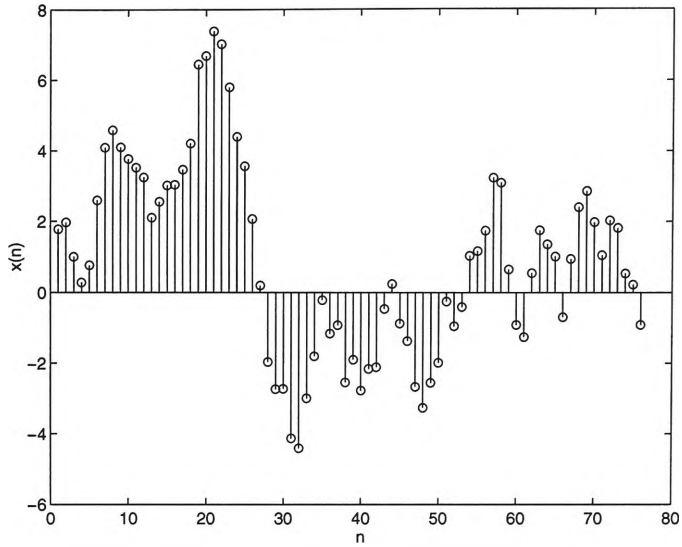

Figure 3.2: An *AR(1)* signal with r=0.96

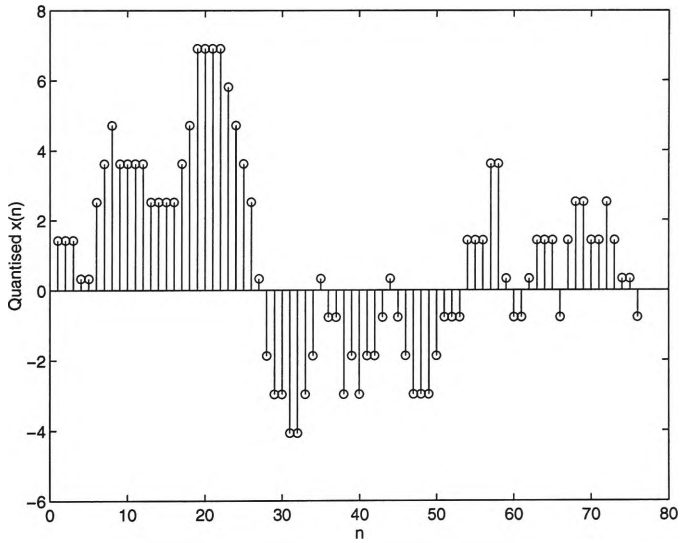

Figure 3.3: 4 bpp quantised version of the signal shown in Figure 3.2

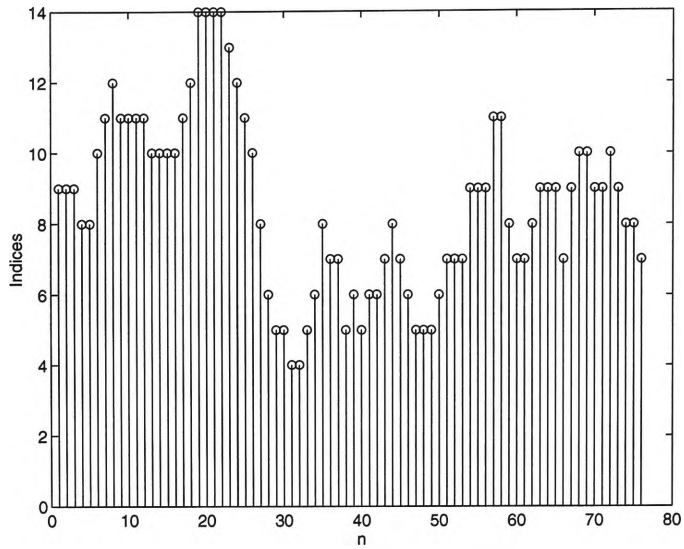

Figure 3.4: Indices of 4 bpp quantised version of the signal shown in Figure 3.2

3.2.2 Pixel and index domains correlation in vector quantisation

This section investigates the relationship between the pixel and index domains correlation for the indices obtained from vector quantising natural images. This investigation has been performed for FSVQ and TSVQ, and carried out on two sample images displayed in Figures 3.5 and 3.6.

Figure 3.5: *Lena*

Figure 3.6: *Airplane*

3.2.2.1 Pixel and index domains correlation in TSVQ

A tree can be considered as a map that represents a family generation in a way that every node of the tree is a descendant of the corresponding node in the

level before it. The tree map can give the relationship between any two leaves or nodes of the tree. In TSVQ, the indices of vector quantised images are the labels of the tree leaves. They represent the path from a leaf to the root of the tree, and this path shows a tree leaf's ancestors. In other words, these labels are not only the symbols to find a match for a vector from a look-up table, but also express the genealogy of each vector. Hence the more the similarity between two vectors, the more similar will be their corresponding indices.

High correlation in natural images lead to adjacent pixels with about similar intensity, consequently there is a high probability of low MSE difference between adjacent image blocks. This characteristic results in mapping neighbouring blocks to similar codevectors when MSE is used as a matching criterion. In TSVQ, the children of a node have the least possible MSE difference or, small MSE difference between two codevectors lead to their being closer in the tree map, and consequently having more similar indices. The outcome of these observations is that the indices of a tree structured vector quantised image are highly correlated.

A universal binary TSVQ codebook, with the rate of 8 bpv, was generated to demonstrate the above argument. The indices of two test images, shown in Figures 3.5 and 3.6, are obtained. Two tests were performed to show the relationship between the pixel and index domain correlation. The correlation in the pixel and index domain were computed, then the possibility of image reconstruction with only the indices is illustrated.

Figures 3.7 and 3.8 respectively demonstrate the row- and column-wise correlation of pixels and the indices. The method of correlation computation for images is as follows; several samples of rows and columns for each image have been randomly selected, and the average value of the correlation in each direction has been utilised as an estimation of correlation. Of course for non-stationary sources such as images, the correlation is not constant all over the image, for example the correlation in the areas containing edge or texture is less than in the low activity areas such as background. The results presented only give an estimation of the correlation under the assumption of a stationary image model [Jain 1989, pages 189-209]. The outcome of results is that the first lag correlation of pixels and indices are very similar.

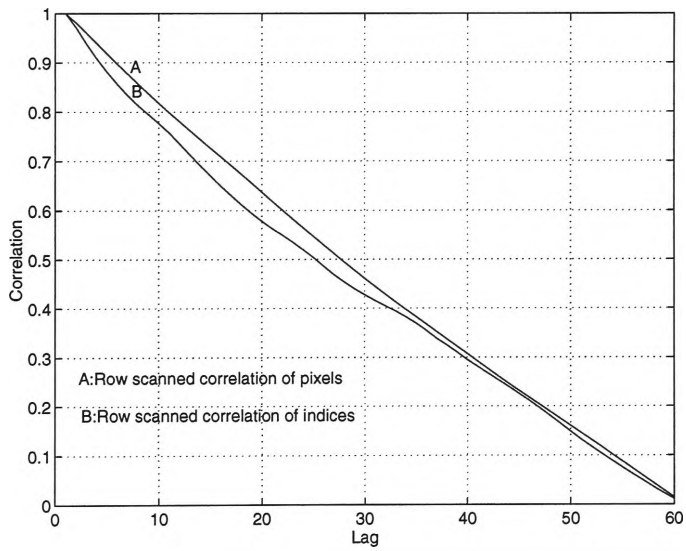

Figure 3.7.a: Row-wise correlation of image's pixels and indices for the image "*Lena*"

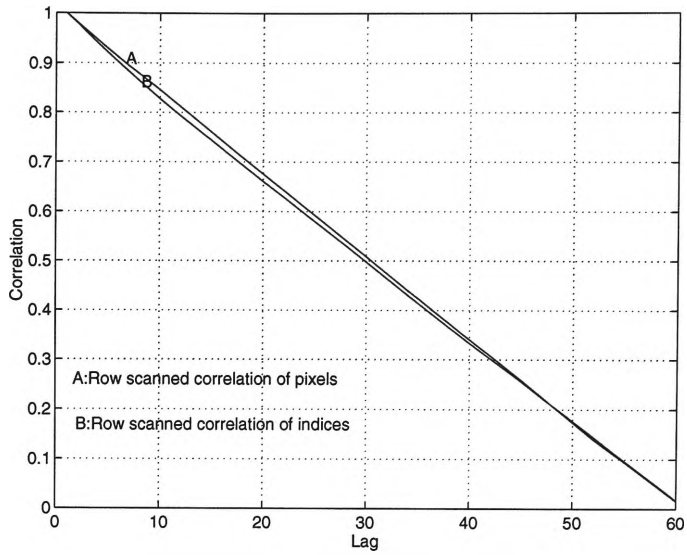

Figure 3.7.b: Row-wise correlation of image's pixels and indices for the image "*Airplane*"

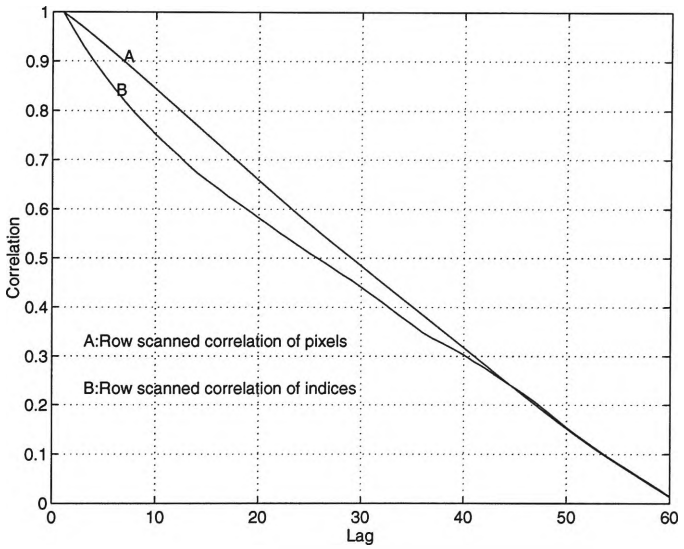

Figure 3.8.a: Column-wise correlation of image's pixels and indices for the image "*Lena*"

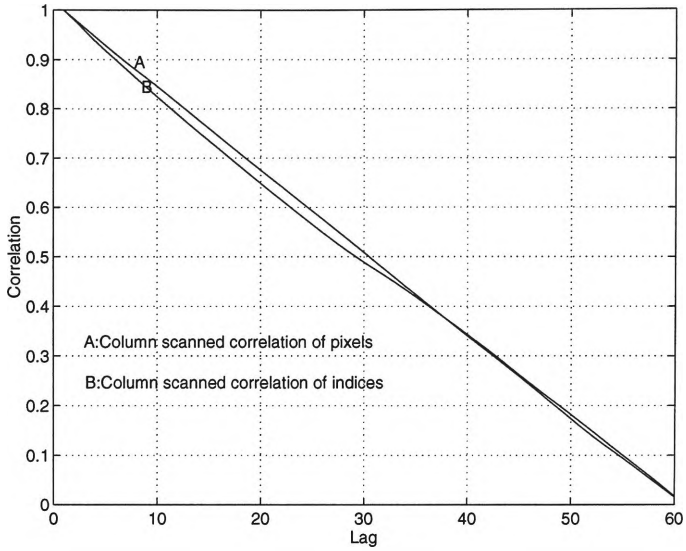

Figure 3.8.b: Column-wise correlation of image's pixels and indices for the image "Airplane"

It is possible to reconstruct an image from its indices. Figures 3.9 and 3.10 illustrate the reconstructed version of two images based on their indices. The image is tree-structured vector quantisation with average rate of 6 bpv. A constant value was added to the indices to show the images more clearly. This experiment is another empirical proof for the relationship between the index and pixel domains correlation.

Figure 3.9: *Lena* by indices

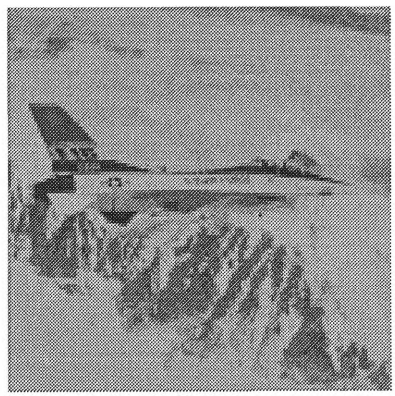

Figure 3.10: *Airplane* by indices

3.2.2.2 Pixel and index domains correlation in full-search vector quantisation

In a FSVQ, directly extending the relationship between the pixel and index domains correlation, as in TSVQ, is difficult. This is because of the differences between the methods of codebook generation, and index assignment of these two schemes. The indices in FSVQ are just symbols employed as a key to reconstruct a block and convey no other information as in TSVQ. Figures 3.11 and 3.12 illustrate the pixel and index domains correlation of a FSVQ scheme (8 bpv) respectively. The graphs labelled "A" and "B" show the correlation in the pixel and index domains respectively. It is evident that the index domain correlation is different from the pixel domain correlation.

In order to see the existence of the analogy between the inter-block and inter-pixel correlation in FSVQ and TSVQ, the codevectors have been ordered based on their energies. Of course, this may not be the best method of ordering the codevectors to evaluate the inter-indices correlation, but it is sufficient to show the analogy between the index and pixel domains correlation, and also the method is computationally efficient. The graph labelled "C" in Figures 3.11 and 3.12 illustrate the index domain correlation after codevectors re-ordering. The relationship between the correlation in pixel and index domains is evident from these figures. The first lag correlation of indices and pixels in both cases, row- or column-wise, are very similar to each other.

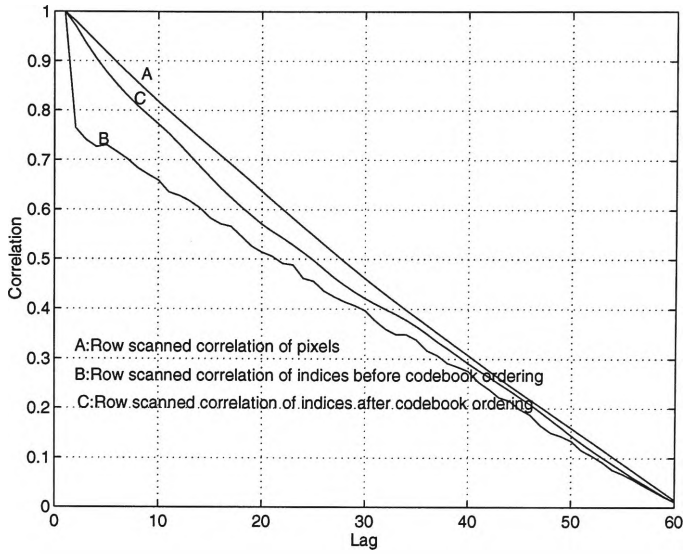

Figure 3.11.a: The row-wise correlation of indices and pixels for the image "*Lena*"

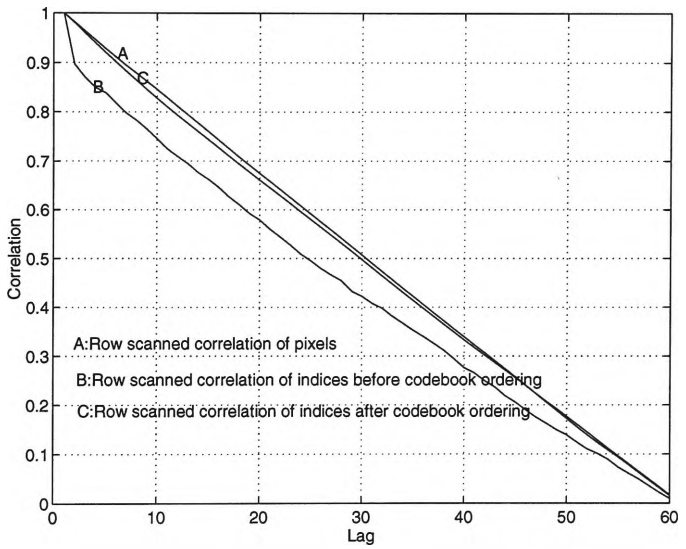

Figure 3.11.b: The row-wise correlation of indices and pixels for the image "*Airplane*"

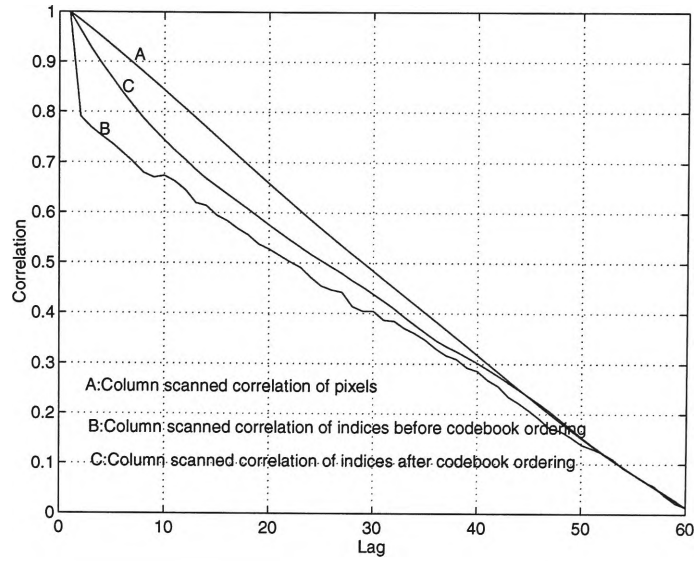

Figure 3.12.a: The column-wise correlation of pixels and indices of image "*Lena*"

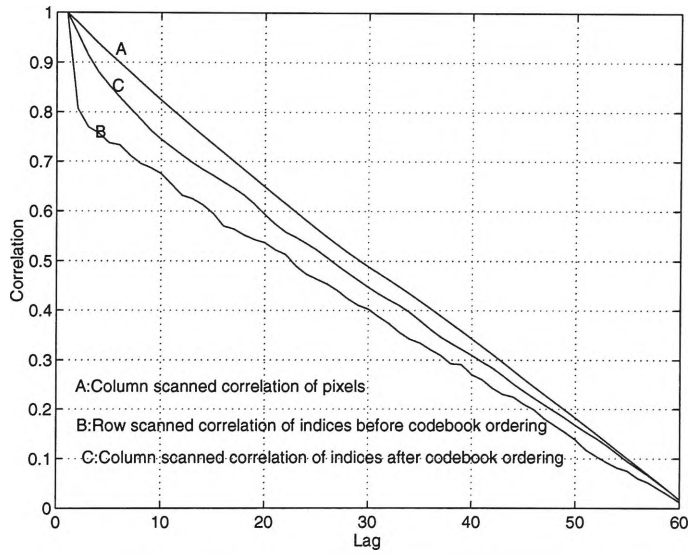

Figure 3.12.b: The column-wise correlation of pixels and indices of image "*Lena*"

3.3 The Probablity of Identically Indexed Neighbouring Pixels or Blocks

An effect of scalar quantisation on highly correlated sources was the possibility of mapping neighbouring pixels onto identical cells (Figures 3.2 and 3.3). If two neighbouring pixels have the same index, transmitting both of them is redundant. Side information to show the map of pixels with identical indices and a fraction of the whole indices are sufficient to reconstruct the image. The

capability of this approach in image compression requires an investigation of the probability of identically indexed neighbouring blocks. This section details an empirical investigation of this probability for scalar and vector quantisation.

3.3.1 The probablity of identically indexed neighbouring pixels in scalar quantisation

The probability of identically indexed neighbouring blocks has been measured for *AR(1)* sources. Figure 3.13 shows this probability for quantisations from 1 to 8 bpp. This probability for 2 bpp when r=0.99 is about 0.95, thus implying that the information of only five percent of pixels and the information of the map of identically indexed neighbouring pixels are sufficient to reconstruct the signal.

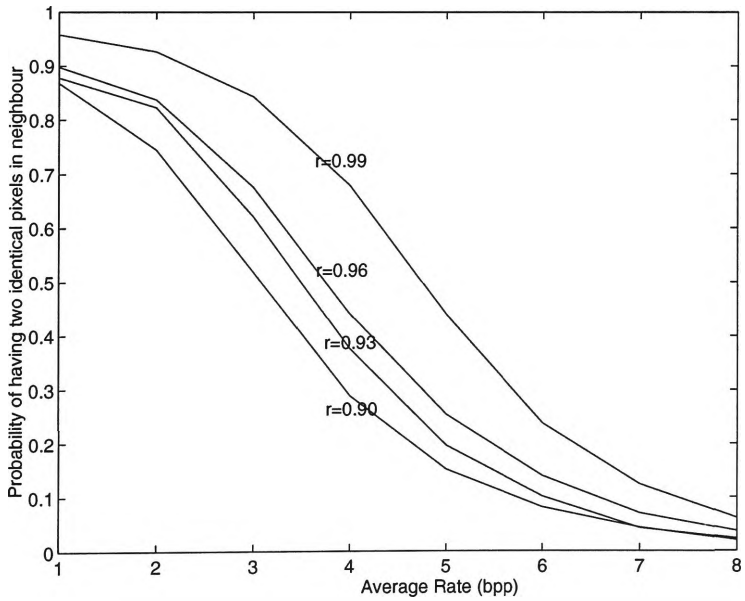

Figure 3.13: The probability of having neighbouring pixels with identical indices after quantisation

3.3.2 The probablity of identically indexed neighbouring blocks in vector quantisation

This section investigates the probability of identically indexed neighbouring blocks when the image is vector quantised by FSVQ or TSVQ. An experiment was performed by employing a set of 26 images from the USC database to show this probability. The average rate of the codebooks in this experiment was 1 to 7 bpv.

Six groups of probabilities were measured. These are the probabilities that a block has a neighbour, located in the north-west-side, north-side, west-side, north- or west-side, north-west-, north- or west-side, or north-west, north-east, north- or west-side, with an identical index with the block index. A, B, C, D, E and F are respectively used to show these six cases for simplicity in presentation .

Appendix A gives the results obtained for all these cases, for 2x2 and 4x4 block sizes. Figures 3.15 and 3.16 show the average results of all 26 images, when the block size is 4x4. The results indicate that the probability of identically indexed neighbours is a function of three variables, bit rate, block size and number of neighbouring blocks. This probability is an increasing function of the number of neighbouring blocks and a decreasing function of block size and bit rate.

The probability results for TSVQ are higher than those for FSVQ. The reason is the search constraint in TSVQ. The blocks have more choice in a FSVQ than TSVQ in finding the best matching codevector. TSVQ restricts each block from all the possible choices by the search constrain. In other words, TSVQ quantises

each block by a smaller codebook than FSVQ. Consequently this results in having more identically indexed neighbouring blocks for TSVQ than FSVQ.

The effect of block size is such that the smaller block size increases the probability of identically indexed neighbouring blocks. The high correlation in natural images results in low pixel variation in small areas, and consequently neighbouring small blocks tend to have pixels with about the same intensity. Hence, more neighbouring blocks are mapped onto the same codevector, when compared with the situation where the block size is large.

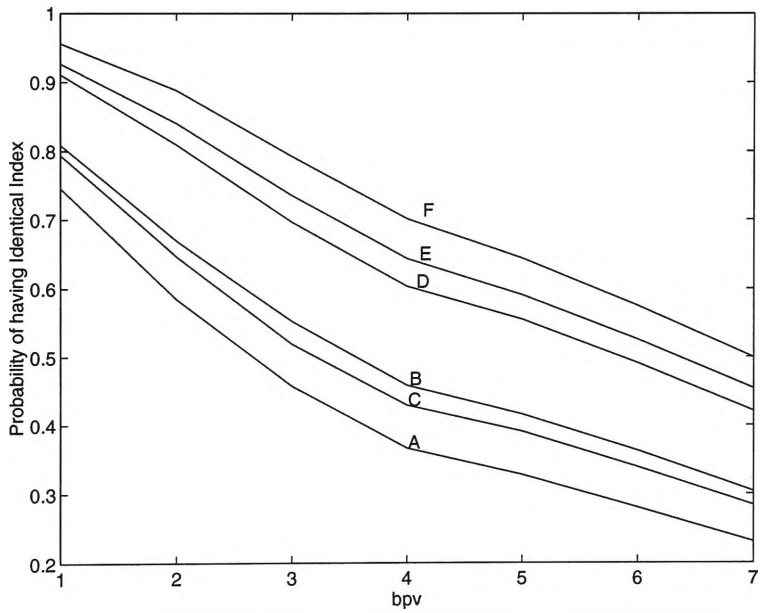

Figure 3.14: The probability results of having an identical index with neighbouring blocks in six cases for indices obtained from a TSVQ scheme

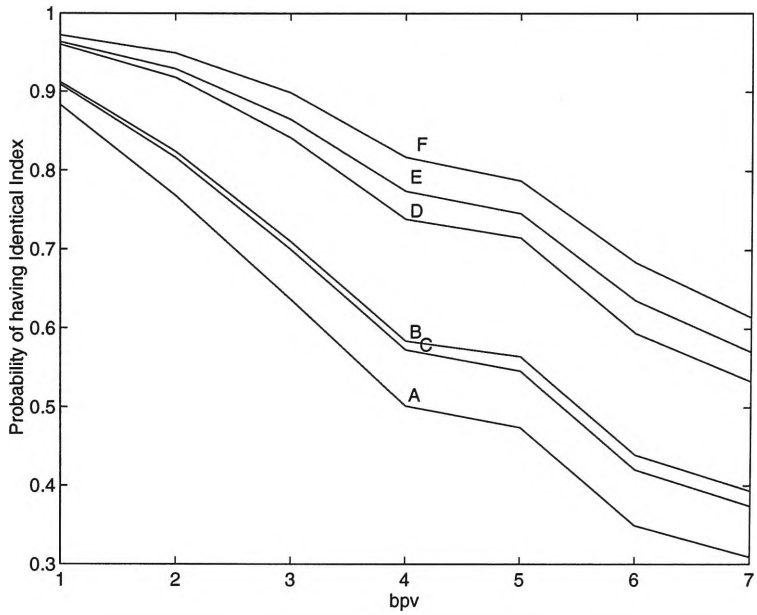

Figure 3.15: The probability of having an identical index with neighbouring blocks in six cases for indices obtained from a FSVQ scheme

3.4 Summary

This chapter investigated two characteristics of indices obtained from scalar and vector quantisation for a source model and some natural images. The first section of this chapter presented the relationship between the index and pixel domain correlations. It was shown that the indices obtained from the quantised version of highly correlated sources were highly correlated as well. The second section showed the existence of a high probability of having identically indexed neighbouring blocks for highly correlated sources such as images. This probability increases as the block size decreases.

Chapter 4

Index Compressed VQ Method I

4.1 Introduction

Chapter 3 showed that the probability of identically indexed neighbouring blocks is high. This characteristic derives from the high correlation among the indices of image blocks. Here, a simple method is introduced to exploit this characteristic in order to improve the performance of simple VQ schemes such as FSVQ and TSVQ. The possibility of having two neighbouring blocks with identical indices obviates the need to transmit or store block indices in all cases.

Two sets of information are required to avoid the transmission of redundant information: a map indicative the identically indexed neighbouring blocks' location and another showing those indices that cannot be recovered from the knowledge of the index of their neighbouring blocks. Since this method compresses the indices obtained by a VQ scheme, and it is the first method

introduced in this thesis, it is called index compressed VQ method I (IC-VQ (I)).

The rest of this chapter is organised as follows. The next section introduces IC-VQ (I). Section 4.3 derives the bit rate formulae of IC-VQ (I), and Sections 4.4 analyses the performance of IC-VQ (I). The final section presents the simulation results, and compares this new method with other coding techniques.

4.2 Index Compressed VQ Method I

The indices of image blocks, in the traditional VQ schemes, are transmitted without considering their dependency on the indices of their neighbouring blocks. Here, a new method, IC-VQ (I), is introduced which is able to exploit the indices' dependency. The procedure entailed is expounded through an example.

5	5	5	5	5	5	5	5	7	7	7	7	7	3	3	3
5	5	5	5	5	5	7	7	7	7	7	7	3	3	3	3
5	5	5	5	4	4	4	4	4	1	1	1	1	1	1	1
5	5	3	3	4	4	4	4	4	4	4	4	1	1	1	1
2	2	2	2	2	2	2	2	8	8	8	8	8	8	8	8
5	3	3	3	4	4	4	4	1	1	1	1	1	a	a	a
c	c	c	c	c	c	c	b	b	b	b	b	b	b	b	b
c	c	c	c	c	b	b	b	b	b	b	9	9	9	9	9
c	c	b	b	b	b	b	b	9	9	9	9	9	9	9	9
c	c	c	b	b	b	9	9	d	d	d	d	d	d	d	d
c	b	b	9	9	9	d	d	d	d	d	d	d	d	d	d
f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f
r	r	r	r	c	c	c	b	b	b	b	b	b	b	b	b
d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d
e	e	e	e	e	e	e	e	e	e	e	e	e	e	e	e
o	o	o	o	o	o	o	n	n	n	n	n	q	q	q	q

Table 4.1: Sample matrix of blocks indices

Table 4.1 shows a schematic example of the indices of a quantised image. In Table 4.1, it can be easily seen that instead of transmitting all the indices,

only needs transmit or store the index of a block and the address location of those blocks which are identically indexed. Here, the address location is called the map of identical indices (MII). Table 4.2 shows the MII generated when comparing the indices in Table 4.1 in a row scanned fashion. A symbol "1" shows that a block has an identical index with the preceding block in the same row, and a symbol "0" shows that the index of the block has to be transmitted, because its corresponding index cannot be recovered from the preceding information, or there is no preceding information.

0	1	1	1	1	1	1	1	0	1	1	1	1	0	1	1
0	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1
0	1	1	1	0	1	1	1	1	0	1	1	1	1	1	1
0	1	0	1	0	1	1	1	1	1	1	1	0	1	1	1
0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	0	1	1	1	1	0	1	1
0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
0	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1
0	1	0	1	1	1	1	1	0	1	1	1	1	1	1	1
0	1	1	0	1	1	0	1	0	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1

Table 4.2: Result of blocks' indices after finding identical indices in row scanned

In Table 4.2 the index dependency in one direction has been exploited, however the probability results from Chapter 3 show that considering more neighbouring blocks increases the probability of finding blocks having neighbours with an identical index. This will result in less information transmission for the indices, and is the basis of considering the column index dependency as well. In considering the dependency along the columns, one only needs to test those blocks that failed the row-wise dependency test, and another MII has to be generated.

Table 4.3 shows the MII resulting from the column dependency test. Three symbols are used, a blank, "1", and "0". A blank shows that a block has an identical index with its west-side neighbouring block and there is no need to transmit any information about it, because it can be completely recovered from its preceding index. A symbol "1" shows that a block has an identical index with its north-side neighbouring block, and a symbol "0" shows that the block has no identical index with its west or north side neighbouring blocks. The indices of blocks labelled as "0" need to be transmitted.

0								0					0		
1							0					0			
1					0					0					
1			0		1							0			
0								0							
1		0			0				0					0	
1								0							
1						0						0			
1			0						0						
0				0			0		0						
0		0			0										
0															
0															
0															
0															
0															
0															

Table 4.3: Result of blocks' indices after finding identical indices in column scanned fashion

4.3 The rate formula of Index compressed VQ (I)

The rate of IC-VQ (I) consists of two parts; the required rate for the MII information, and the rate of indices to be transmitted. Thus, the rate can be written as:

$$R = R_{MII} + R_{indices} \quad \text{bpv} \tag{4.1}$$

The combined MII information consists of three symbols as pointed out earlier and the rate, which is required to transmit the MII information, can be computed as follows. Let P_W be the probability that a block has an identical

index with its west-side neighbour. Then $(1\text{-}P_W)$ is the probability of occurrence of the rest of the block types. R_{MII} consists of 1 bpv to represent the neighbouring blocks with an identical index in the row direction, and $(1\text{-}P_W)$ bpv, to represent the neighbouring blocks with an identical index in the column direction, and the rest of the blocks. Thus,

$$R_{MII} = 2 - P_W \quad \text{bpv} \tag{4.2}$$

The number of indices to be transmitted can be found by subtracting the number of blocks which have neighbours with an identical index in their north or west side from all the image's blocks. This number corresponds to the probability of having blocks with an identically indexed neighbours. The table for this probability is given in Appendix A, and here the probability itself is indicated by $P_{W\cup N}$. The average rate of indices to be transmitted is:

$$R_{indices} = r(1 - P_{(W\cup N)}) \text{ bpv} \tag{4.3}$$

where r is the rate of the basic VQ scheme in its original form. The final rate can be written as:

$$R = 2 - P_W + r(1 - P_{(W\cup N)}) \text{ bpv} \tag{4.4}$$

Note that (4.4) is based on considering two neighbours located in the north and west. It is possible to consider more than two neighbours and gain a reduction in the rate of the indices. The price of this gain is the need for extra information to represent MII. It is difficult to conclude that considering more than two neighbours will result in an overall performance improvement without further analysis. The case of three neighbours is next considered. Using the same method of deriving (4.4) it can be shown that the rate when three neighbours are considered is:

$$R = 3\text{-}P_W - P_{(W\cup N)} + r(1\text{-}P_3) \text{ bpv} \tag{4.5}$$

where P_3 is the probability of having identically indexed neighbouring blocks when three neighbours are considered. The difference between (4.4) and (4.5) can be written as:

$$difference = 1 - P_{W \cup N} + r(P_{(W \cup N)} - P_3) \quad \text{bpv} \tag{4.6}$$

The situation where considering three neighbours results in better improvement requires having a difference less than zero. This results in the following inequality:

$$r > \frac{1 - P_{(W \cup N)}}{P_3 - P_{(W \cup N)}} \tag{4.7}$$

The results included in Appendix A shows that the difference between $P_{(W \cup N)}$ and P_3 is marginal. This implies that a better coding performance can be obtained only when 3 neighbours are considered at very high bit rates. However, at very high rates, firstly the VQ encoding process will be computationaly too expensive, because of the codebook size, and secondly all the probabilities discussed in Chapter 3 are very small, and consequently based on (4.4) and (4.5) IC-VQ (I) may have a lower performance than its corresponding VQ scheme.

4.4 Performance Analysis of IC-VQ (I)

This section shows the situation where the rate obtained through (4.4) is less than the rate of the corresponding VQ. If the rate of a basic VQ is r, then to have $R \leq r$ requires:

$$2 - P_W + r(1 - P_{(W \cup N)}) \leq r \tag{4.8}$$

After some straight forward manipulations, inequality (4.5) can be written as:

$$\frac{2 - P_W}{P_{(W \cup N)}} \leq r \tag{4.9}$$

As $P_{(W \cup N)}$ is the probability of having an identical index with any of the neighbouring blocks in the north- or west-side, it can be written as:

$$P_{(W \cup N)} = P_W + P_N - P_{(W \cap N)} \tag{4.10}$$

where $P_{(W \cap N)}$ is the probability of having blocks with an identical index with both neighbouring blocks located in north- and west-side at the same time. On the basis of the simulation results presented in Appendix A it can be seen that,

$$P_W \approx P_N \tag{4.11}$$

This results in,

$$\frac{\frac{2}{P_W} - 1}{2 - \frac{P_{(W \cap N)}}{P_W}} \leq r \tag{4.12}$$

In the worst case,

$$P_{(W \cap N)} = P_W \tag{4.13}$$

In the best case,

$$P_{(W \cap N)} = 0 \tag{4.14}$$

The following are results for the worst and best cases respectively,

$$P_W \geq \frac{2}{r+1} \tag{4.15}$$

$$P_W \geq \frac{2}{2r+1} \tag{4.16}$$

Table 4.4 shows the results for the possibility of P_W to being greater than $\frac{2}{r+1}$ for bit rates from 1 to 7 bpv and block size 2x2 and 4x4 for all the 26 test images.

The results show that for rates from 5 bpv to 2 bpv the proposed scheme always outperforms the traditional VQ schemes. For 1 bpv the performance is always lower, because the required rate for the MII information alone is more than the actual rate based on the traditional VQ schemes. For a 1 bpv quantiser this method requires some modification to exploit the correlation in the MII information. There are only two possibilities for the indices of blocks: the symbols "0" or "1". Consequently, the MII information consists of runs of either of these two symbols that can be easily

exploited. Another approach for the 1 bpp case is to encode the indices directly by run-length coding without using the general approach of IC-VQ (I), since at 1 bpv quantisation, there exist a run of identical indices as can be seen from table of results in Appendix A. At rates more than 5 bpv the probability of getting better results than traditional VQ schemes is very high (96%). When P_W is compared with $\frac{2}{2r+1}$, the results show that the probability of improvement is always unity.

	Block Size	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
TSVQ	2x2	0.96	0.96	0.96	1.00	1.00	1.00	0.00
FSVQ	2x2	0.96	0.96	1.00	1.00	1.00	1.00	0.00
TSVQ	4x4	0.96	0.96	1.00	1.00	1.00	1.00	0.00
FSVQ	4x4	0.96	0.96	0.96	1.00	1.00	1.00	0.00

Table 4.4: The probability of having better results than traditional VQ schemes

4.4.1 Maximum performance improvement of IC-VQ (I) over basic VQ

To find out the maximum improvement of IC-VQ (I) over the corresponding VQ, the maximum difference between the rate of these two coders at a given distortion has to be found . This difference can be obtained by subtracting the rate of these two coders. This formula for the case when the basic VQ has been used is given in (4.17).

$$R_{diff} = rP_{(W \cup N)} - 2 + P_W \tag{4.17}$$

where R_{diff} is the difference rate. The percentage of bit rate improvement can be obtained from (4.18).

$$Percentage\ of\ improvement = \left(\frac{R_{diff}}{r}\right) x100 \tag{4.18}$$

For the average values of $P_{W \cup N}$ and P_W and block size 4x4 pixels the results obtained by (4.17) and (4.18) are shown in Figures 4.1 and 4.2 respectively. It can be seen that the actual bit rate improvement at higher rates for IC-VQ (I) is more than at lower rates, and the percentage improvement, at bit rates from 5 to 3 bpv, is more than the rest.

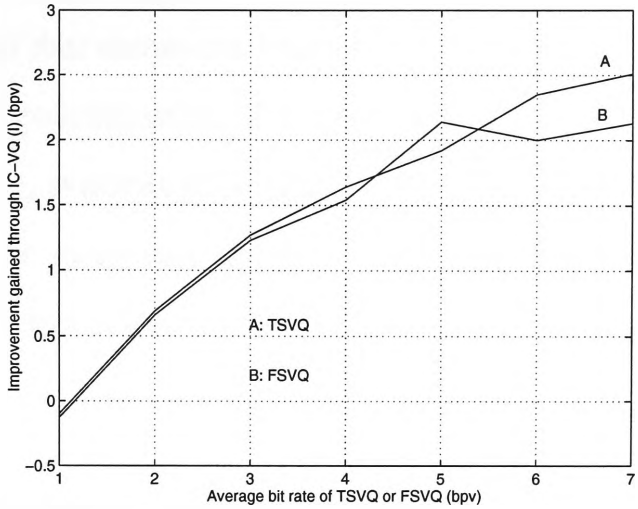

Figure 4.1: Actual bit rate improvement of new scheme over traditional VQ coders (tested image "*Lena*")

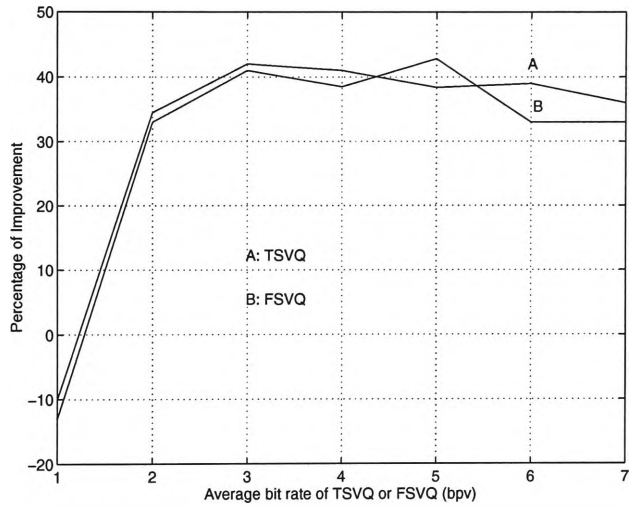

Figure 4.2: Percentage of bit rate improvement of IC-VQ (I) over basic VQ (tested image "*Lena*")

4.4.2 Methods for improving the performance of IC-VQ (I)

The MII information generates an overhead that needs to be transmitted. This section describes a method of reducing this overhead based on the structure of the MII. The fact that blocks with identically indexed neighbour are highly probable results in identical adjacent symbols in the MII. An obvious method to reduce this rate is to exploit this feature by run-length coding [Jayant 1984, Chapter 10].

In run-length coding instead of transmitting several identical adjacent symbols a number that shows the frequency of their repetition is transmitted. For example if a concatenation of a row scanned version of the symbols in Tables 4.2 and 4.3, symbols of "0" and "1" that shows MII information, is to be transmitted, the generated streams of symbols can be divided into two sets. These two sets show the repetition of the symbols "1", and "0" respectively.

These sets can be separately Huffman coded [Huffman 1952]. For a sample test image, *Lena*, the sets of frequency of ones and zeros are shown in Figures 4.3.a and 4.3.b respectively. It can be seen that the possibility of having a repetition of ones or zeros up to sixty exists, and instead of transmitting several identical bits only a few bits that show the repetition is enough. This results in considerable bit saving.

	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
obtained by equation 4.4	7.77	6.43	5.04	3.87	2.59	1.74	1.26
With Run-Length coding the MII	6.51	5.36	4.21	3.23	2.14	1.32	0.70

Table 4.5: The rate for the worst case with and without run-length coding the MII information (image *Baboon*)

Among all the tested images, the only one that gives a lower performance, when the results of IC-VQ (I) are compared with the basic VQ schemes, was *Baboon*. The result for the image *Baboon* was obtained by transmitting the MII information with run-length coding to see the effect. Table 4.5 shows the results in two cases; with run-length coding the MII information and the one obtained by equation (4.4) and for 4x4 block size and FSVQ scheme. It can be seen that when the MII information are run-length coded, the new scheme always outperforms FSVQ. The same results have been achieved for the TSVQ as well.

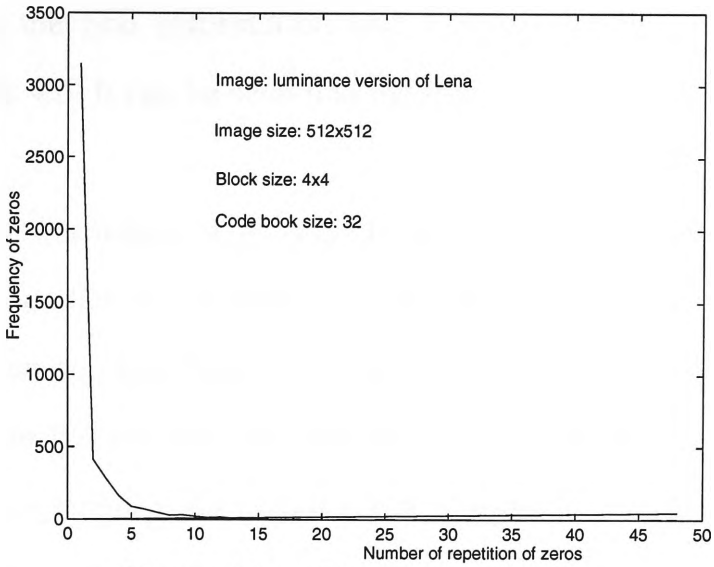

Figure 4.3.a: Frequency of zeros

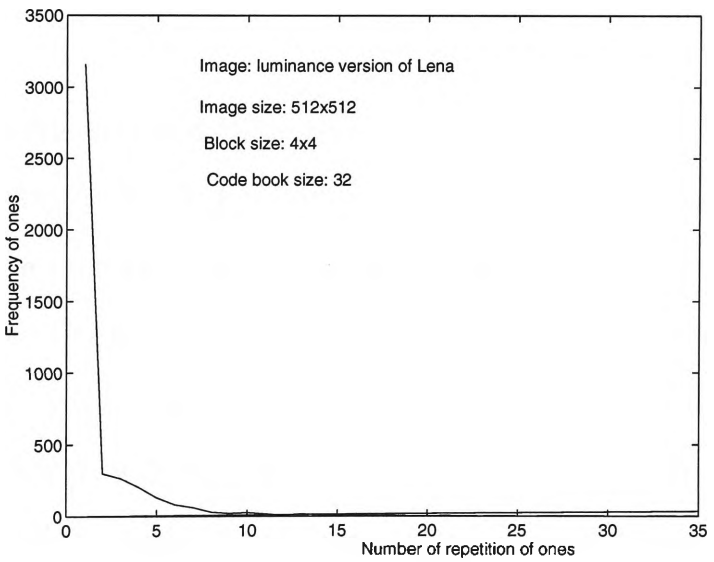

Figure 4.3.b: Frequency of ones

Bit rate of FSVQ	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
Bit rate of IC-VQ (I)	4.317	3.483	2.438	2.123	1.430	0.864	0.421
percentage of improvement	38	42	51	47	52	57	58

Table 4.6: The final results of IC-VQ (I) with run-length coding the MII information and Huffman coding the indices and comparison with FSVQ

Another step to improve the performance is to apply a lossless coding, such as Huffman coding, on the indices. The obtained rate and the percentage of

improvement for FSVQ and 4x4 block size for the image *Lena*, after run-length coding the MII information and Huffman coding the indices, are shown in Table 4.6. It can be seen that the minimum improvement is 38%.

One of the disadvantages of IC-VQ (I) was its low performance at 1 bpv. The reason is that the rate to represent MII information is always more than 1 bpv. This can be seen from (4.2). In (4.2) P_W is always less than 1, this leads to the rate for the MII information being more than 1. However, the utilization of run-length coding on the MII information changes the situation. For all the 26 test images this results in having a bit rate less than 1 bpv for the MII information.

4.5 Simulation Results and Discussion

The performance of IC-VQ (I) based on two VQ schemes, FSVQ and TSVQ, has been compared with the corresponding VQ coders in two cases; with and without Huffman coding of the indices of the VQ coders. The other comparison has been made between the IC-VQ (I) and the JPEG coding standard [Wallace 1992] and address VQ. The comparison with JPEG has been performed because of two reasons. First, JPEG belongs to the family of transform coding techniques, and this comparison shows the advantage or disadvantage of IC-VQ (I) in terms of objective measurement in comparison with a sophisticated transform coding scheme. Secondly JPEG is a standard coding scheme and some researchers provide results by comparing their techniques with JPEG, and here this comparison gives the opportunity to evaluate IC-VQ (I) or other techniques introduced in this thesis. These comparisons are performed on the luminance version of *Lena*. The image and block sizes are 512x512 and 4x4 pixels respectively. The PSNR has been computed by the following formula:

$$PSNR = 10\log_{10}\left(\frac{255^2}{mse^2}\right) \tag{4.19}$$

4.5.1 Comparison between IC-VQ (I) based on FSVQ and FSVQ

Figures 4.4 shows the results of comparing IC-VQ (I) based on FSVQ and FSVQ at bit rates ranging from 0.05 bpp to 0.35 bpp. A similar comparison is shown in Figure 4.5 for IC-VQ (I) based on FSVQ and FSVQ; in this case the indices of FSVQ are Huffman encoded. The improvement acheived by the new scheme over FSVQ is more than 3 dB on average (Figure 4.4). This improvement is more significant at bit rates from about 0.06 to 0.2 bpp which correspond to 0.125 to 0.3125 bpp for FSVQ. About the same results were anticipated earlier in section 4.4. It can be seen in Figure 4.5 that about the same improvement as in Figure 4.4 can be obtained at low bit rates, but at bit rates more than 0.2 bpp the difference is about 2.7 dB while in the previous case it was about 3.5 dB. The reason adduced for the better performance of IC-VQ (I) over a combination of FSVQ (with Huffman coding of the indices) is that both FSVQ and Huffman coder disregard the indices dependency.

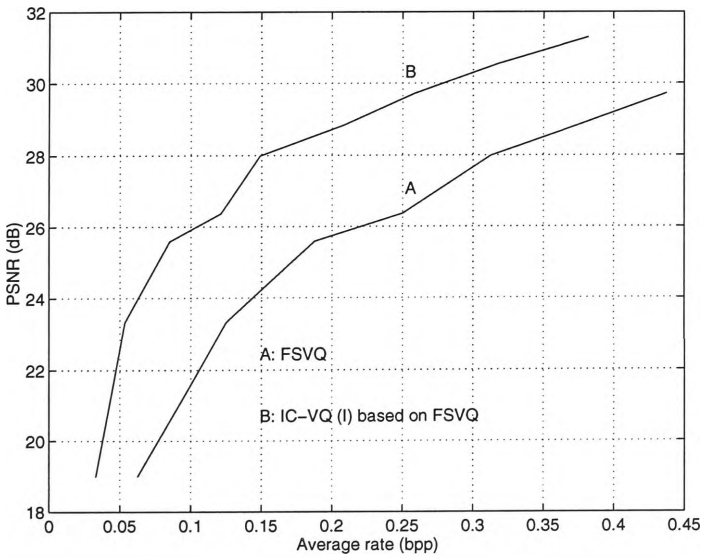

Figure 4.4: The results of FSVQ and IC-VQ (I)

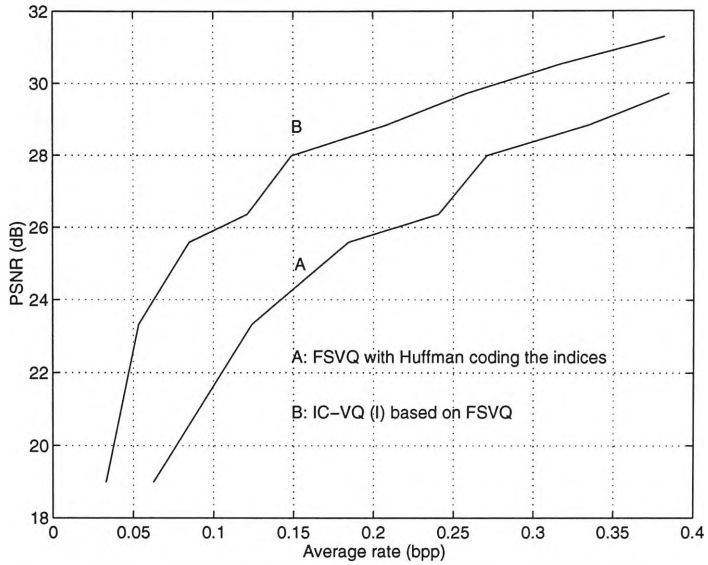

Figure 4.5: The results of FSVQ with Huffman coding the indices and IC-VQ (I)

4.5.2 Comparison between IC-VQ (I) based on FSVQ and JPEG

In Figure 4.6 the results of IC-VQ (I) based on FSVQ scheme, and JPEG are shown. This scheme significantly outperforms JPEG at bit rates less than 0.2 bpp. The PSNR difference at 0.15 bpp is about 3 dB. The reason for this improvement is that JPEG (or transform coding techniques) at low bit rates follows the same process as at high rates. Lack of bits at low rates results in attention just to a few transform domain coefficients, neglecting the effect of other coefficients on the coded image quality, and ignoring the inter-block correlation. IC-VQ (I) exploits the inter-block correlation, which is neglected by JPEG, to improve its performance at low bit rates.

However, there is a penalty that IC-VQ (I) has to pay for better performance over JPEG, and its counterpart VQ at low bit rates. That is its sensitivity to channel noise. If any bit from two groups of information in IC-VQ (I) changes, the noise will propagate through the rest of the image. This is not the case for JPEG, or the basic VQ, because each block is coded separately, and any change to the stream of bits of one block affects that block only and does not result in noise propagation.

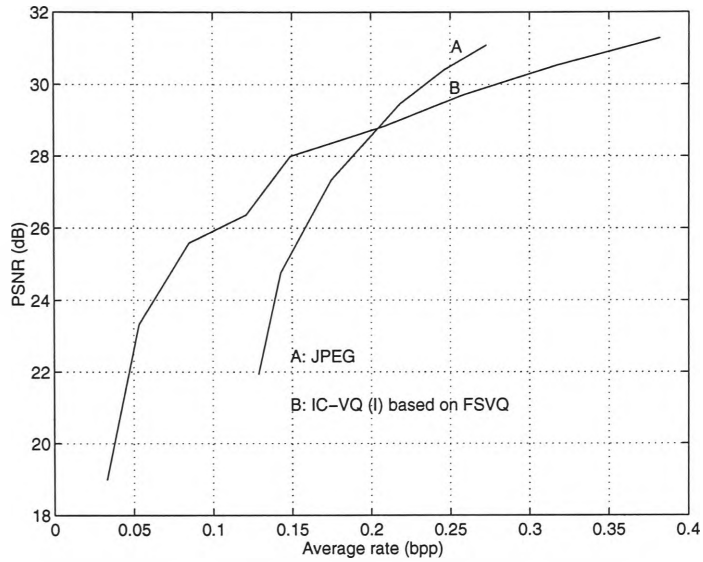

Figure 4.6: The results of JPEG coding standard and IC-VQ (I) based on FSVQ

The other factor in the comparison is the simplicity of implementation. The bit rate has no effect on the implementation of JPEG, while this factor significantly reduces the computational time of FSVQ during encoding. The decoder part of the new scheme is only a look-up table and a Huffman decoder, while the JPEG decoder requires a DCT transformer with hardware or software, Huffman decoding (or arithmetic decoding), and a dequantiser. It should be noted that at bit rates more than 0.2 bpp the PSNR of JPEG is higher than that of the new coder.

4.5.3 Comparison between IC-VQ (I) and TSVQ

In Figures 4.7 and 4.8 the results of IC-VQ (I) based on TSVQ, and TSVQ with the indices Huffman encoded are shown. The improvement acheived by the new scheme over TSVQ is about 2 dB on average (Figure 4.7) which is less than the results obtained by applying the new scheme on the indices of FSVQ.

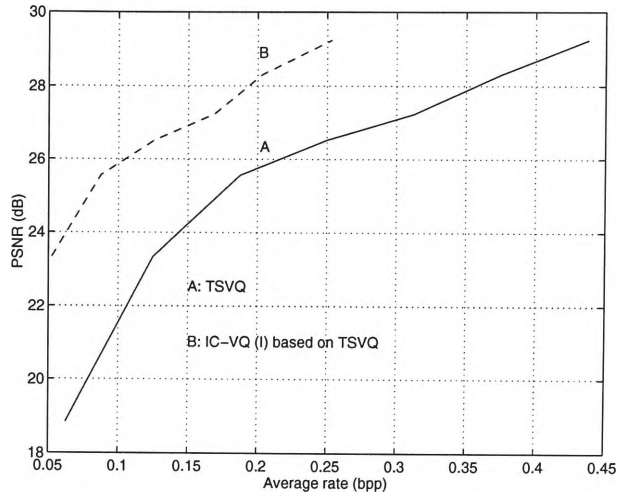

Figure 4.7: The results of TSVQ and IC-VQ (I) based on TSVQ

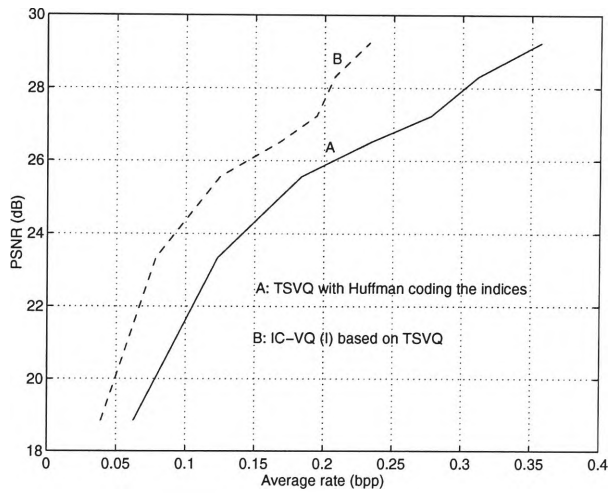

Figure 4.8: The results of TSVQ with Huffman encoding the indices and IC-VQ (I) based on TSVQ

The reason for this difference is investigated by considering the rate of indices to be transmitted and the rate of MII information. For both schemes, FSVQ and TSVQ, the rates are shown in Figures 4.9 and 4.10 respectively. The rate of MII information for both cases is about the same. However the rates of indices to be transmitted are different. This rate, after Huffman encoding, for FSVQ is less than it is for TSVQ. The probability of identically indexed neighbouring blocks for a TSVQ is higher than for a FSVQ, and this results in more uniformity in the frequency of indices to be transmitted. This leads to rendering Huffman encoding (or entropy encoding) of the indices less effective.

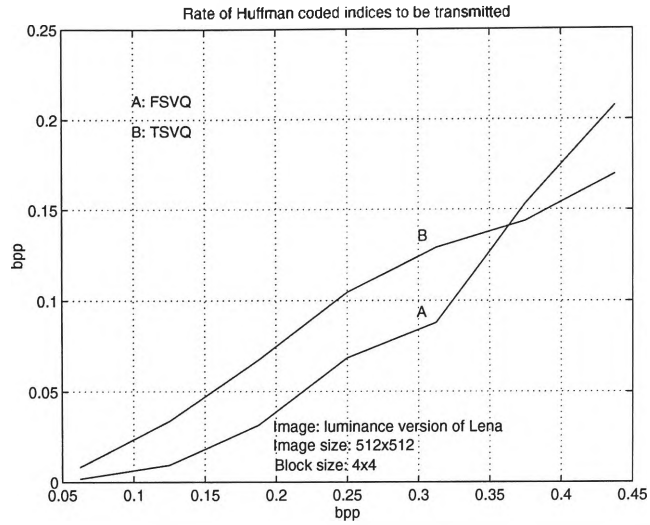

Figure 4.9: Rate of Huffman coded indices to be transmitted

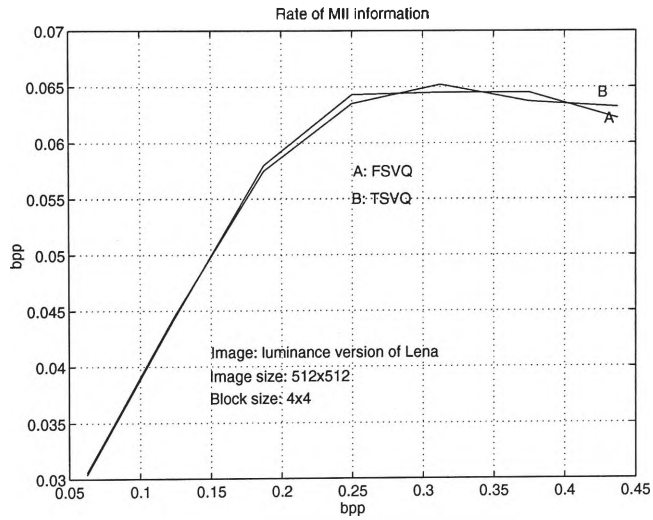

Figure 4.10: Rate of MII information

4.5.4 Comparison between IC-VQ (I) based on TSVQ and JPEG

In Figure 4.11 the results of the new scheme based on TSVQ is compared with JPEG. The performance of the new scheme outperforms JPEG at bit rates less than 0.16 bpp. At 0.16 bpp the PSNR is about 26.5 dB.

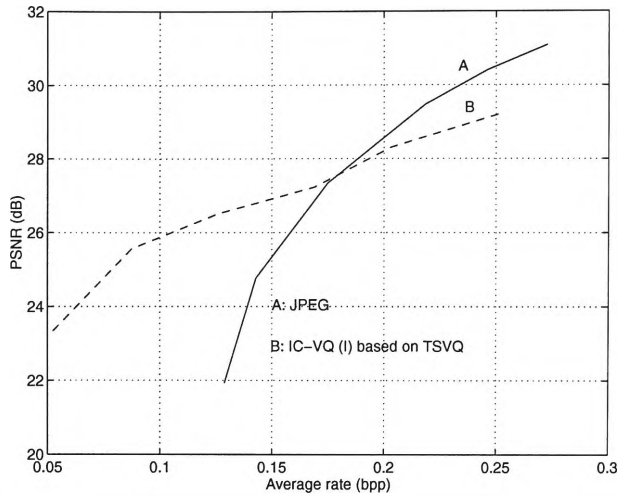

Figure 4.11: Comparing TSVQ by IC-VQ (I) with JPEG

4.5.5 Comparison between IC-VQ (I) and address VQ

A simple comparison can show the superiority of this scheme over address VQ in terms of implementation. This scheme consists of a simple memoryless VQ encoder and a comparator to find the identically indexed neighbours. These requirements are much less than the complexity of address VQ which was discussed in Chapter 2. Another important advantage of this system is its flexibility to operate on a wide range of compression ratios using a small amount of space for the codebook of the memoryless VQ. While in the case of address VQ any change of the compression ratio requires not only some memory space for the memoryless VQ codebook but also a high amount of memory space for the address codebook, and if the time required to generate the address codebook is considered the advantage of IC-VQ (I) over address VQ becomes obvious. The simulation results show about 0.9 dB difference in PSNR at about the same rate (0.26 bpp).

4.6 Summary

This chapter proposed a novel VQ technique with superior performance than traditional VQ schemes. The coder exploits the indices' correlation to improve the performance of the vector quantisation schemes. The traditional VQ schemes transmit or store the index of each image block without

considering its surrounding blocks, thus neglecting the existing inter-block correlation.

The new scheme exploits the inter-block correlation by using the indices' characteristics at low bit rates. In this method, the image block indices' information is divided into two sets: first a map that shows whether a block has an identical index with its neighbouring blocks, and secondly the index of those blocks that do not have an identical index with their neighbours. These two sets of information are enough to reconstruct the image.

It was analytically and experimentally shown that the new scheme outperforms the traditional VQ coders. The performance of the new scheme when applied to FSVQ shows better improvement than when applied to TSVQ. At bit rates less than 0.2 bpp the new scheme outperforms the JPEG coding standard, while requiring much less complexity than JPEG coding standard in implementation. However, IC-VQ (I) is more sensitive to channel noise than the basic VQ and JPEG. The proposed coding scheme outperforms address VQ in terms of implementation simplicity and memory space requirements, but gives a lower PSNR at about the same rate.

Chapter 5

Index Compressed VQ Method II

5.1 Introduction

Chapter 4 presented a simple index compression method, IC-VQ (I), based on the characteristics of identically indexed neighbouring blocks. IC-VQ (I) exploites the inter-block correlation on the basis of the fact that at low bit rates the probability of having neighbouring blocks with identical indices is high. Two problems associated with the algorithm are: its application when the probability of identically indexed neighbouring blocks is low, and the fact that the characteristics of the identically indexed neighbouring blocks is not the only indices' feature that can be used in inter-block correlation removal.

The probability of identically indexed neighbouring blocks is a decreasing function of the rate, and this feature makes IC-VQ (I) less effective as the average rate of the basic VQ coders increases. Each pair of the neighbouring blocks has some correlation and they may not necessarily have an identical index after quantisation. This chapter introduces another scheme, index compressed VQ method II (IC-VQ (II)), for exploiting this correlation.

The theoretical basis of IC-VQ (II) stems from a consequence of the high indices' correlation in natural images; neighbouring image blocks have a high probability of being mapped to highly correlated codevectors. In other words, VQ maps the neighbouring blocks onto a very small subset of the

codebook containing codevectors that are relatively highly correlated. If one considers a small subset of codebook instead of the whole codebook, it is possible to improve the performance of traditional VQ schemes. In IC-VQ (II)the image blocks undergo a vector quantisation step, and the generated indices are sent through a compression step by mapping them onto a subset of the indices. This approach leads to assigning less bits than the original VQ coder to each image block.

IC-VQ (II) can be considered as a FS-VQ, where each incoming image block index introduces a subset of the big universal codebook to the encoder or decoder to find the match for the next coming index; in other words each index introduces the next state. In reality there is no need to generate the small codebooks for each of the indices to find out the index of its neighbouring blocks, as it is the case in FS-VQ, where state codebooks has to be generated. IC-VQ (II) directly employs the value of each block index to introduce the small codebook for the neighbouring blocks. This technique does not restrict the encoder to search for the best match among the codebook associated with each index as is the case in FS-VQ. This gives the possibility of improved performance in the sense of finding the best match, but results in a longer search time. The other main difference between FS-VQ and IC-VQ (II) is the method of codebook design. IC-VQ (II) only uses the codebook of a memoryless VQ coder, and its design procedure is as simple as a memoryless VQ, which is much simpler than designing a FS-VQ. The design of a FS-VQ involves an initial classifier, state space, next state function and state codebook design, and also requires iterative state codebook improvement [Gersho 1992].

IC-VQ (II) much like the previously introduced IC-VQ (I) consists of the combination of a VQ coder and a lossless coding scheme to compress the image block indices. The index compression procedure in IC-VQ (II) is such that a shorter index derived from a subset of the original VQ codebook is transmitted. From this point of view this method is similar to the recent work presented as a variant of address VQ [Cheng 1995]. In this variant of address VQ, besides the codebook for the memoryless VQ coder, several big codebooks have to be designed to encode the indices losslessly. While this scheme outperforms address VQ, its encoder is more complex than the original address VQ. In IC-VQ (II), like address VQ, the encoding of the indices requires a codebook, but this codebook is a subset of the codebook of the original VQ coder and does not need to be generated separately. Thus IC-VQ (II) saves memory space by performing all the encoding process with one codebook, and requires less complexity when compared with the second and third steps of index compression in address VQ.

The rest of this chapter is organised as follows. Section 5.2 shows that neighbouring blocks' indices belong to a small subset of the VQ codebook. Section 5.3 details the new method of index storage (or transmission). This section gives the rate formula of IC-VQ(II) based on pure rate without considering any other coding scheme to compress the output data. Because this is the case in basic VQ. Section 5.4 considers the performance of the new coding method. Section 5.5 presents and discusses the results of simulations, and Section 5.6 gives the summary.

5.2 Neighbouring Image Blocks' Indices Characteristics

Chapter 3 showed that the probability of identically indexed neighbouring blocks at low bit rates was high. For the case of TSVQ a high correlation

exists among the indices, and for the case of FSVQ when the codevectors of the codebook are reordered the same phenomenon can be observed easily. A high correlation among the image block indices indicates that the probability of a given block having neighbours with about the same index is high. This is because in VQ, each image block is mapped onto the codevector which has the best similarity, in terms of a specific criterion such as MSE. Consequently the neighbouring image blocks' indices are mapped onto similar codevectors which construct a small subset of the codebook.

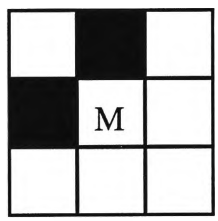

Figure 5.1: The indices of black coloured blocks are compared with the index of block M

A simple experiment was conducted to reveal the relationship among the indices of neighbouring blocks. Consider the image block arrangement shown in Figure 5.1; the block under consideration is labelled "M" and the considered neighbours are the black coloured blocks to its north and west. The frequency of occurrence of the neighbouring blocks indices is computed for each block. Figure 5.2 shows a plot of the frequency (vertical axis) versus the image block indices and, the index of the neighbours. This figure is based on testing the indices of neighbouring blocks using four standard images from the USC database, and the indices are obtained by full search vector quantising the images when the codevectors of the codebook are re-ordered based on their energies. The block size for this experiment is 4x4 pixels and the codebook size is 128. The result shown in Figure 5.2 is an empirical evidence that the probability of mapping neighbouring blocks onto a small subset of the VQ codebook is very high.

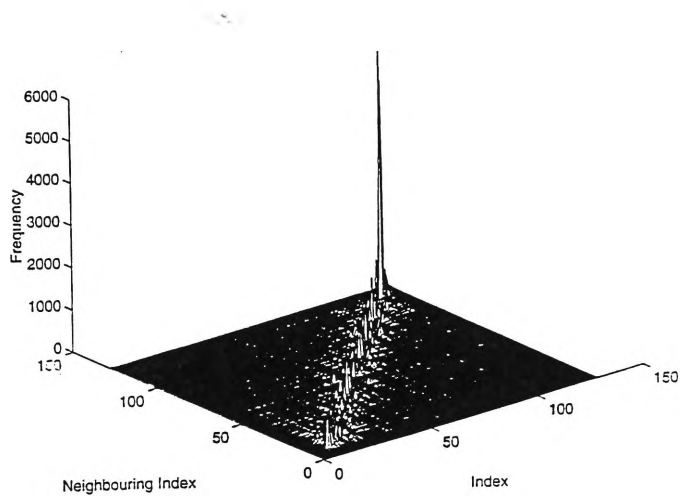

Figure 5.2: Frequency of the indices of the neighbouring blocks with a block which has a specific index

5.3 Index Compressed VQ Method II

The characteristics of indices shown in Section 5.2 can be used to reduce the coding rate by transmitting or storing a special index (probabilistic index) which describes the neighbourhood situation of a block. In the following consideration, the VQ codebook has N codevectors and the VQ rate is $r=log_2N$. Figure 5.2 shows that the neighbouring blocks are mapped onto a small subset of the VQ codebook, so there is a need for only a few symbols instead of N to represent the neighbourhood situation of an image block. These symbols are referred to as probabilistic indices.

The indices of blocks in the neighbourhood of a given block can be classified as any of four groups: (i) indices with identically indexed west-side neighbour, (ii) indices with identically indexed north-side neighbour (iii) indices whose west-side or north-side neighbour is differently indexed.

The last category is further divided into those indices whose west- or north-side neighbours either have a high or low probability of occurrence.

The first step of IC-VQ (II) is similar to IC-VQ (I) which exploits the feature of identically indexed neighbouring blocks. The probability that a block has neighbours with the same index is much higher than otherwise at low bit rates; this is evident from Figure 5.2, and the results presented in Appendix A. If the blocks' indices in this situation are identified, one only needs to assign the probabilistic index to the rest. The index of each block is compared with its west-side neighbour (i.e. the index of block *M* is compared with that of the block labelled "*W*" in Figure 5.3), to identify the blocks that have identically indexed west-side neighbour. If such indices are identical, a specific symbol such as "1" is assigned to it, otherwise another symbol such as "0" is assigned. It is clear that the required bit-rate to represent this information is 1 bpv.

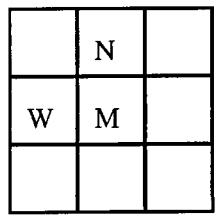

Figure 5.3: In IC-VQ (II), the index of two neighbours in north- and west-side (blocks N and W) is used to encode the index of block M

Next, the index of the remaining blocks (that failed the first test) is compared with that of its north-side neighbour (i.e. the index of block *M* is compared with the index of the block labelled "*N* " in Figure 5.3). The number of blocks in this comparison is $n\overline{P}_w$, where n is the total number of image blocks and $\overline{P}_w$ is the probability that a randomly chosen image block will have an index different from that of its west-side neighbour. Graph labelled "H" in Figure 5.4 shows this probability, for a typical image (*Lena*),

for codebooks having bit rates ranging from 1 to 8 bpv. As in the previous step, a "*1*" is assigned to a block if there is a match, and a "*0*" otherwise. The actual bit rate for this step is $\overline{P}_w$ bpv.

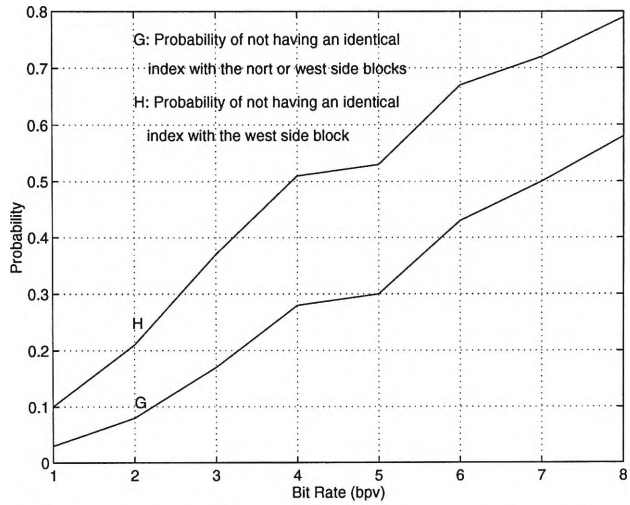

Figure 5.4: $\overline{P}_W$ and $\overline{P}_{(W \cup N)}$ versus rate

The number of remaining blocks is $n\overline{P}_{(W \cup N)}$, where $\overline{P}_{(W \cup N)}$ is the probability that a block does not have an identical index with any of its west- or north-side neighbours. This probability is plotted (graph labelled "G") for rates ranging from 1 to 8 bpv for the image *Lena* in Figure 5.4. The remaining blocks are divided into two groups: those with high probability of having a neighbouring block with about the same index and those with a low probability of occurrence. The division into these two groups is performed in two steps. First, it is identified to which of its neighbour in the west or north position a block is most similar index-wise. A "1" is assigned, if the block is most similar to its west-side neighbour and "0" otherwise. The required rate in this step is $\overline{P}_{(W \cup N)}$ bpv. Next those blocks are identified which have a neighbouring block with an index with very low probability of occurrence. The required rate in this step is $\overline{P}_{(W \cup N)}$ bpv. It is necessary to transmit or store the complete index for these blocks as well. The rate of this case then becomes $(\overline{P}_{(W \cup N)} + rP_{low})$ bpv, where P_{low} is the probability of having a

neighbouring block whose index has a very low probability of occurrence, and r is the rate of a conventional VQ scheme such as FSVQ or TSVQ.

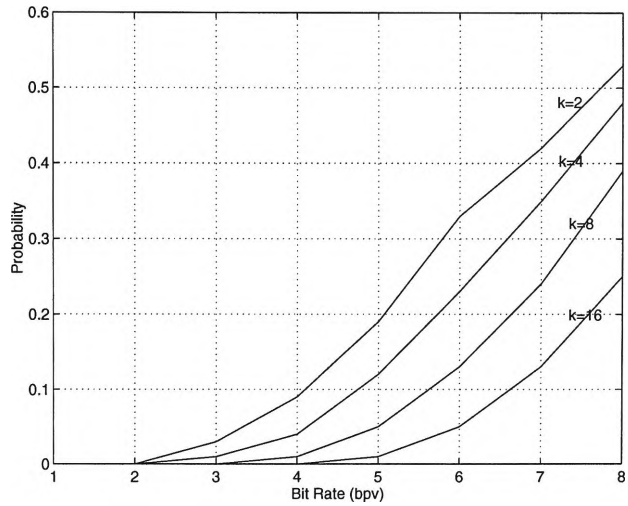

Figure 5.5: The probability that both of the neighbours of a block in west- and north-side has an index with high difference value with the block index

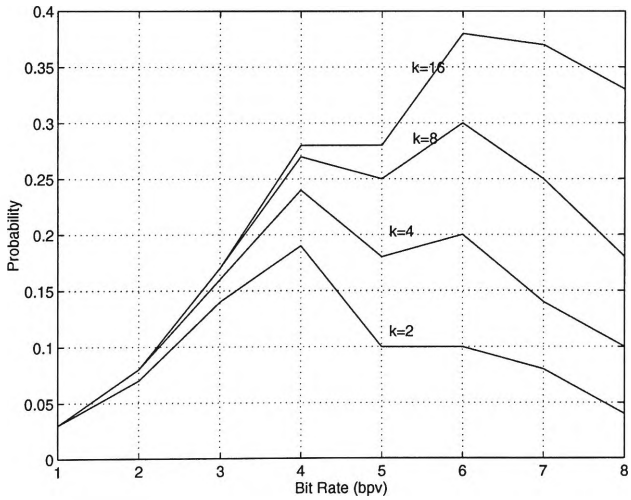

Figure 5.6: The probability that any of two neighbours of a block in west- and north-side has an index with low difference value with the block index

A probabilistic index is assigned to the remaining blocks. The rate of this section depends on the number of probabilistic indices. If the north- or west-side neighbours are used, 1 or 2 bpv has to be assigned for all the remaining blocks. The number of remaining blocks is $n(1-P_{(W\cup N)}-P_{low})$ where $(1-P_{(W\cup N)}-P_{low})$ is the probability that the west- and north-side neighbouring blocks are

neither identical to the given index nor do they belong to the group of blocks with a low probability of occurrence. This probability is referred to as P_{high} . The required rate for this section is $P_{high} log_2 k$ bpv, where k is the number of probabilistic indices. The total rate of this algorithm is:

$$R=1+\overline{P}_W+2\overline{P}_{(W\cup N)}+rP_{low}+P_{high}log_2 k \text{ (bpv)} \quad (5.1)$$

P_{low} and P_{high} depend on r and k. Figures 5.5 and 5.6 illustrate these two probabilities for the image *Lena* respectively.

IC-VQ (II) considers two neighbouring blocks in the compression process. One can consider three or more neighbours. This may not necessarily improves the performance of IC-VQ (II), since in the case of more than two there is a need to transmit or store extra overhead information to introduce the neighbouring block situation.

5.4 Performance Analysis of IC-VQ (II)

This section discusses the optimum situation for IC-VQ (II) , introduces the method of improving the performance of IC-VQ (II) and compares IC-VQ (II) with its corresponding VQ schemes.

5.4.1 Minimum rate of IC-VQ (II)

In (5.1), P_{low} and P_{high} depend on k and r, and the rest are constants or dependent on r. It is possible to find the optimum k (for a given image) by finding the minimum rate for a specific codebook size. Figure 5.7 illustrated the rate versus k for codebooks of sizes varying from 16 to 64 (4 to 6 bpv) and block size 4x4. The optimum k is about 4 for those codebook sizes. In other words for universal codebooks with sizes varying from 16 to 64 a regional codebook with size about 4 gives the minimum rate.

Two cases, (when $k=1$ and $k=2^N$) are noteworthy as there is no need to transmit overhead information about the classes with high and low probability of occurrences. In the two extreme cases, $k=1$ and $k=2^N$, the rate formula can be written as:

$$R=1+\overline{P}_W+2\overline{P}_{(W\cup N)}+rP_{low} \text{ (bpv)} \qquad \text{if } k=1 \tag{5.2}$$

and,

$$R=1+\overline{P}_W+2\overline{P}_{(W\cup N)}+rP_{low}, \text{ (bpv)} \qquad \text{if } k=2^N \tag{5.3}$$

These two cases consist of two classes, those blocks with an identical index with their neighbour and the rest. This consideration respectively results in,

$$\overline{P}_{(W\cup N)}= P_{\text{low}} \tag{5.4}$$

and

$$\overline{P}_{(W\cup N)}= P_{high} \tag{5.5}$$

If the extra overhead information is removed, then the rate formula can be expressed as,

$$R=1+\overline{P}_W+r\overline{P}_{(W\cup N)} \tag{5.6}$$

This formulae is the same as the rate formulae for IC-VQ (I).

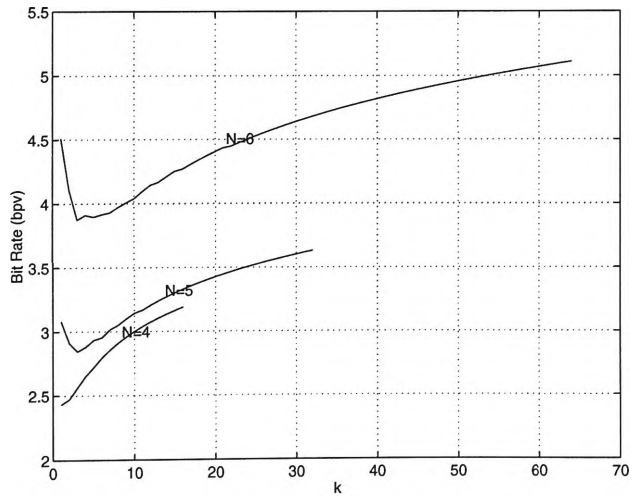

Figure 5.7: Rate versus k

The main difference between the rate formulas given in (5.1) and (5.6) is the requirement of overhead information for the high and low probability classes. Next the condition under which the rate given by (5.6) results in an

improvement is derived. Note that under this condition, the optimum k will be 1 or 2^N. The desired condition is easily obtained by finding the difference between the two formulae.

$$Difference=2\overline{P}_{(W\cup N)}+rP_{low}+P_{high}log_2K-r\overline{P}_{(W\cup N)} \tag{5.7}$$

If rP_{high} is added to both sides followed by some manipulation, (5.7) can be written as,

$$Difference=2\overline{P}_{(W\cup N)}+P_{high}(log_2K-r) \tag{5.8}$$

If the difference in (5.8) is less than zero, then the rate obtained by (5.6) gives a better improvement. The required condition is:

$$r-log_2K<\frac{2\overline{P}_{(W\cup N)}}{P_{high}} \tag{5.9}$$

5.4.2 Methods of improving the rate of IC-VQ (II)

It is possible to reduce the bit rate by run-length coding the first three terms of (5.1), since all these three terms are streams of ones and zeroes. Thus, the possibility of having several identical symbols in succession provides a means of further improving the performance of the new scheme. Figure 5.8 illustrates the results of IC-VQ (II) based on FSVQ with and without run-length coding. The improvement achieved by employing run-length coding is about 0.04 bit/pixel at the same PSNR.

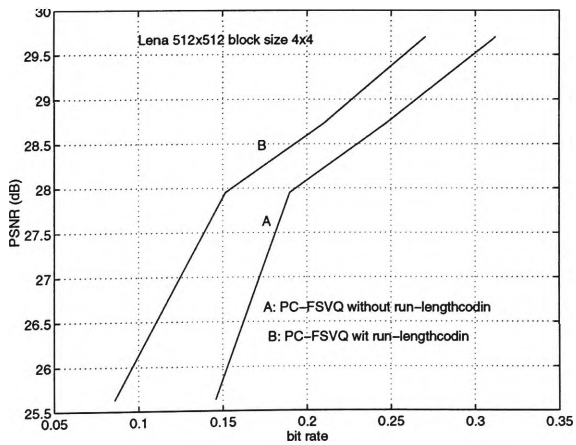

Figure 5.8: Comparing IC-VQ (II) with and without run-length coding

5.5 Simulation Results and Discussion

The coding performance of IC-VQ (II) based on both FSVQ and TSVQ coder is compared at low bit rates with IC-VQ (I), predictive pruned TSVQ (PPTSVQ), JPEG, and finite state VQ (FS-VQ). In most cases the objective performance superiority of the new scheme is demonstrated. The PSNR has been calculated using (4.19).

5.5.1 Comparison between IC-VQ (II) and IC-VQ (I)

Figures 5.9, and 5.10 show the results of IC-VQ (I) (graphs labelled "A") and IC-VQ (II) (graphs labelled "B"). IC-VQ (II) outperforms IC-VQ (I) in both cases, and the performance improvement is about 0.5 dB. The improvement at high bit rates is more as expected. However, IC-VQ (I) is simpler than IC-VQ (II), because only the probability of identically indexed neighbouring blocks is considered in the index compression procedure. In terms of sensitivity to channel noise, both systems are faced with the same problem of error propagation.

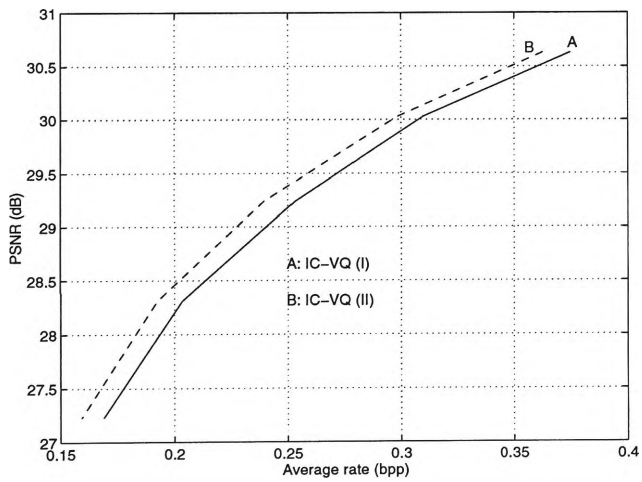

Figure 5.9. Results of IC-VQ (I) and IC-VQ (II) based on TSVQ

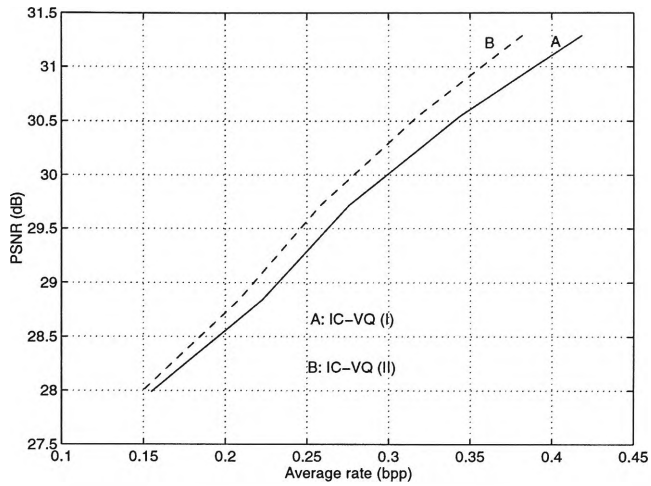

Figure 5.10. Results of IC-VQ (I) and IC-VQ (II) based on FSVQ

5.5.2 Comparison between IC-VQ (II) and predictive VQ

Figure 5.11 presents the results of IC-VQ (II) based on TSVQ and PPTSVQ [Lookabaugh 1993] at low bit rates. IC-VQ (II) always outperforms predictive pruned TSVQ at rates less than 0.25 bpp, and at 0.1 bpp the difference is about 4 dB. A reason for this superiority is that at low bit rates the codebook of the predictive coder is small, hence the active areas cannot be reproduced well, since prediction performs well only in low activity areas. On the other hand, IC-VQ (II), at low bit rates, has a relatively much bigger codebook in comparison with the predictive coder, and consequently is able to reproduce the active areas better.

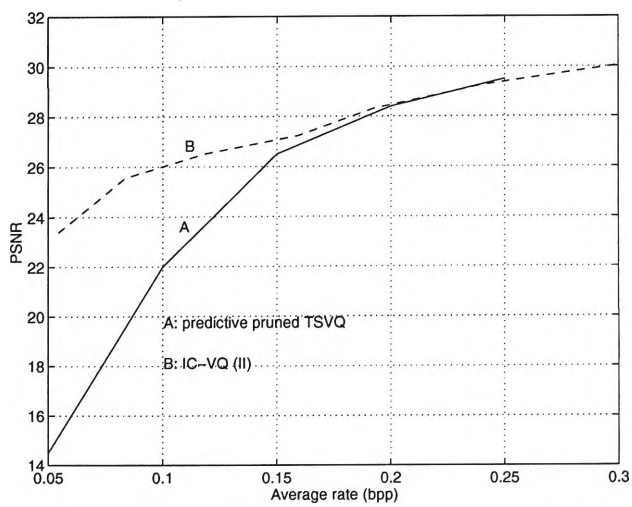

Figure 5.11: Results of PPTSVQ and IC-VQ (II) based on TSVQ

5.5.3 Comparison between IC-VQ (II) and JPEG

Figures 5.12 and 5.13 show the results of JPEG (graphs labelled "A") and IC-VQ (II) based on TSVQ and FSVQ (graphs labelled "B") respectively. The performance of IC-VQ (II) based on TSVQ and FSVQ outperforms JPEG at bit rates less than 0.19 and 0.22 bpp respectively. The performance difference at around 0.15 bpp is about 2 to 3 dB. The reasons for the superiority of IC-VQ (II) are the same as those adduced for IC-VQ (I) in Chapter 4. At low bit rates IC-VQ (II) gains most of its performance improvement through exploiting the inter-block correlation. The quantisation procedure of JPEG adapts the transform domain coefficients based on the low number of bits. A consequence is the similarity of corresponding AC coefficients in neighbouring blocks. This similarity is a source of the quantisation domain inter-block correlation exploited by IC-VQ (I) and (II), but neglected by JPEG. The down side of this is that IC-VQ (II) is more sensitive to channel noise than JPEG.

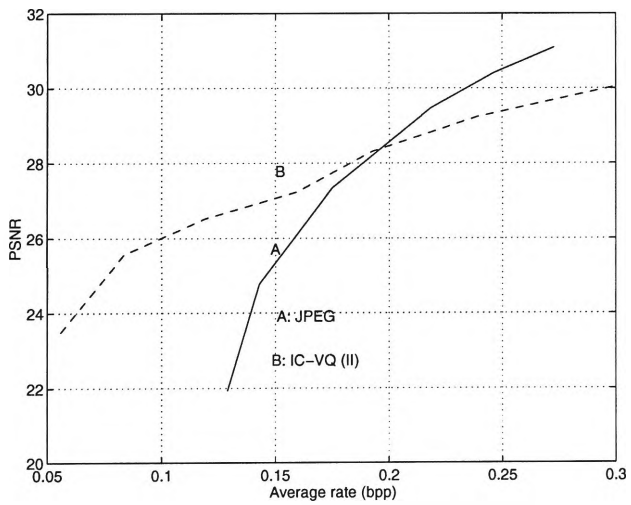

Figure 5.12: Results of JPEG and IC-VQ (II) based on TSVQ

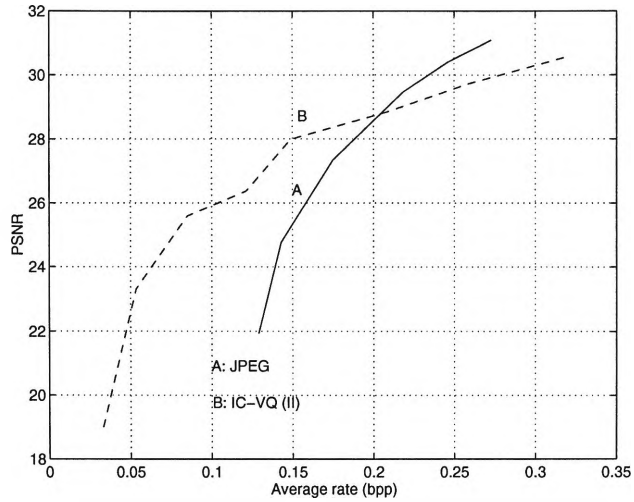

Figure 5.13: Results of JPEG and IC-VQ (II) based on FSVQ

5.5.4 Comparison between IC-VQ (II) and finite state VQ

FS-VQ is able to produce PSNR of 28 dB at 0.25 bpp and 30 dB at 0.25 bpp in conjunction with Huffman coding [Kim 1988]. IC-VQ (II) gives about 29 dB without Huffman encoding the probabilistic indices, the overhead information is however Huffman encoded. The reason for the better performance of IC-VQ (II) over FS-VQ is that, the later exploits the inter-block correlation by a global assumption, while IC-VQ methods are image adaptive in nature. FS-VQ neglects the intra-state dependency, because if two neighbouring blocks have the same state, their indices are transmitted without considering their dependency. In terms of design, IC-VQ (II) requires a simple VQ coder codebook, but the FS-VQ design process, as stated earlier, is very complex.

5.6 Summary

A novel coding scheme, IC-VQ (II), which is capable of providing significant results at low bit rates is proposed. IC-VQ (II) exploits the inter-block correlation by applying a lossless coding scheme on the indices of a vector quantised image to achieve performance improvement. High inter-block correlation in natural images has a direct influence on the probability

of the indices of neighbouring blocks, in a way that their indices belong to a small subset of the codebook. This characteristic has been employed in a lossless scheme to further improve the performance of traditional simple VQ schemes. The coding performance of this scheme has been compared with IC-VQ (I), predictive and finite state VQ schemes, and JPEG at low bit rates. This scheme outperforms these coding schemes.

Chapter 6

Index Compressed VQ Method III

6.1 Introduction

Chapters 4, and 5 presented two index compressed VQ schemes. The aim of both methods is to exploit the inter-block dependency by using the characteristics of the image block indices. IC-VQ (I) employs the feature of identically indexed neighbouring blocks in index compression, but this method ignores the fact that the identically indexed neighbouring blocks are not the only source of inter-block dependency, and also IC-VQ (I) is less effective when the probability of identically indexed neighbouring blocks is low.

IC-VQ (II) resolves the problems of IC-VQ (I) by considering the fact that small areas in natural images require a small codebook for their representation. A problem of IC-VQ (II) is that it assigns a fixed number of bits to areas with varying activities, while more active areas require more bits for their representation than less active areas. In other words, a variable bit assignment procedure depending on the region activity can result in better performance. In this regard, this chapter introduces a method based on TSVQ indices' characteristics, and an extended version for the case where the VQ coder is FSVQ.

The indices obtained by TSVQ have an interesting property that enables them to give information about the correlation between two image blocks. If two image blocks are highly correlated, they may have an identical index, or the same ancestors; for example identical parents, or grand-parents depending on the quantisation levels. The existence of high inter-block correlation in natural images results in having neighbouring blocks with the same genealogy. In other words the neighbouring blocks of a TSVQ quantised image might have the same predecessor up to a particular stage of the codebook tree map. This characteristic can be used to compress the indices, since if the indices of two neighbouring blocks belong to the same generation, the common part of their indices need not be transmitted for both of them.

This chapter introduces a method, index compressed VQ method III (IC-VQ (III)), to exploit the genealogical relation between the image block indices obtained from a TSVQ. IC-VQ (III) is based on the fact that neighbouring blocks mostly belong to the same family, and there is no need to transmit or store the family identification for all of them. This means that IC-VQ (III)

partitions the image into groups of neighbouring blocks belonging to the same family on the basis of the TSVQ codebook.

TSVQ codebook can be considered as a union of a group of small TSVQ codebooks (subtree codebooks). The small subtree codebooks require less bits to represent their members. If the indices of some neighbouring blocks are the children of a subtree, some bit saving can be acheived by indicating the sub-tree and transmitting or storing the common part of the indices. The amount of bit saving is about as much as the difference between the average rate of the original TSVQ tree and the subtree plus some extra overhead to represent the set of neighbouring blocks.

This approach leads to a variable rate bit assignment, because low activity areas contain highly correlated blocks, and high activity areas contain blocks with low correlation. Consequently, the quantised version of blocks from low activity areas are more likely to have an identical ancestry than the blocks from high activity areas. This means that the low activity blocks require a smaller subtree for their representation than the high activity blocks. Therefore, the indices of blocks from low activity areas can be reconstructed with less bits when compared with the indices of blocks from high activity areas.

The basic idea of IC-VQ (III) is similar to variable rate VQ, where the bits are allocated to blocks depending on their activity. The differences are the image adaptivity of IC-VQ (III), and the codebook of the VQ used in IC-VQ (III) is fixed rate. The variable bit assignment procedure of IC-VQ (III) is borne out of considering the characteristic of the image block indices in the encoding procedure, rather than designing a universal-based codebook

variable rate VQ, such as pruned TSVQ [Lookabaugh 1993] or greedy tree growing TSVQ [Riskin 1991]. IC-VQ (III) first assigns bits uniformly, then this uniform bit allocation is changed into a variable one based on the image block location characteristic. A block located in a busy area requires more bits than a block located in a smooth area of the image. Of course, blocks located in active areas of natural images are normally active. Previous variable rate coders allocate bits to each image block based on its activity and disregard the characteristics of the neighbouring blocks.

The differences between IC-VQ (III) and the previous schemes introduced in Chapters 4 and 5 and, address VQ can be considered from the viewpoint of the size of the codebook subset that the indices of the neighbouring image blocks are mapped onto it. IC-VQ (III) maps the neighbouring blocks onto the variable size subsets of the VQ codebook. IC-VQ (I) and (II) map the neighbouring blocks onto fixed size subsets, and address VQ is a hybrid method.

IC-VQ (III) maps the neighbouring image blocks onto the subtrees (subsets) of the VQ codebook depending on the area's activity with some extra side information to represent the subtree. A variable size subtree for image areas results in a variable rate. It can be said that IC-VQ (III) belongs to that category of index compression schemes where a variable size codebook is used to encode the neighbouring blocks (or to represent the indices).

IC-VQ (I) and (II) belong to the category of VQ coders in which the small codebook used to encode the neighbouring blocks is fixed size. In IC-VQ (I) the size of the codebook is one, because the inter-block correlation removal

is based on the identically indexed neighbours, and in IC-VQ (II) the optimum size of the codebook depends on the quantisation levels.

Address VQ can be considered as a hybrid method, where a fixed size codebook has been used to encode the blocks' indices of low activity areas, and a variable size for the rest. Those areas, where the blocks have highly correlated indices, are assigned a fixed size codebook. These areas are naturally low activity areas of the image. The rest of the areas are assigned more bits than the first group. Address VQ has been mainly designed in a hybrid bit allocation form, not because of bit rate efficiency, but because of implementation problems such as codebook design, memory and search.

The rest of this chapter is organised as follows. Section 6.2 presents the genealogical characteristics of indices that have not been covered in Chapter 3, and Section 6.3 introduces the method of index transmission or storage based on the indices' genealogical features. Sections 6.4 and 6.5 give the rate formulae for IC-VQ (III) and methods to improve its compression ratio respectively. Section 6.6 introduces the extended version of IC-VQ (III) for FSVQ, and Section 6.7 presents the simulation results and discussion.

6.2 Genealogical Characteristics of Indices Obtained from TSVQ

Chapter 3 presented the probability of identically indexed neighbouring blocks. In the case of TSVQ, the probability of having neighbouring blocks with identical ancestors (genealogical probability) can be found as well, because the information of neighbouring blocks for an r bpv quantised image shows the probability of having neighbouring blocks with an identical ancestry for $r+1$ bpv, $r+2$ bpv, and so on.

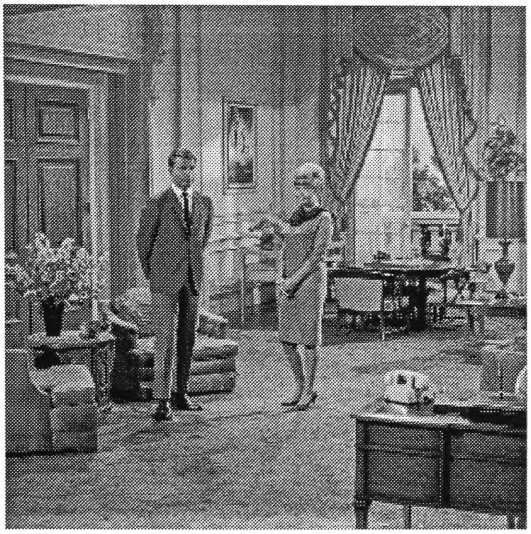

Figure 6.1: Couple

Figure 6.2 illustrates the genealogical index probabilities for a typical image (*Couple*). In that figure, Graph A shows the genealogical probability of having identical ancestors with the block in the north-side, and Graph B shows the genealogical probability of having identical ancestors with the blocks in the north- or west-side. For example for a 7 bpv TSVQ, the results of 7, 6 and 5 bpv show the probability that the index of two neighbouring blocks can have an identical index, parents, and grandparents respectively. A complete result for a set of standard images is given in appendix A for block sizes of 2x2 and 4x4 pixels.

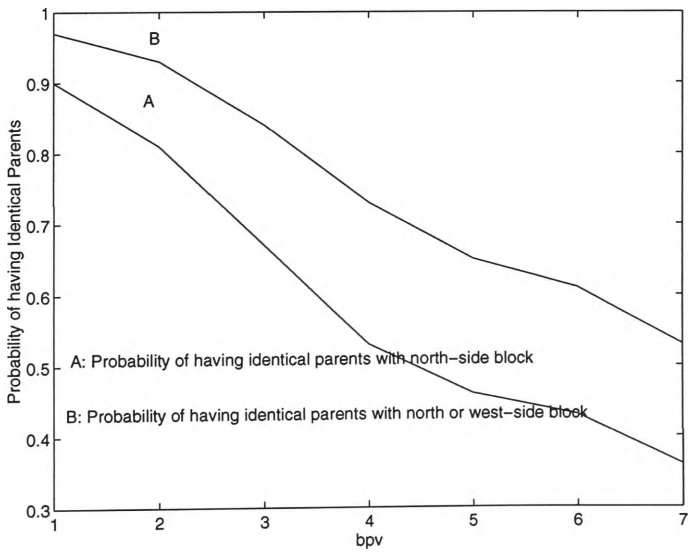

Figure 6.2: The probability of having blocks with identical ancestry

It can be seen that if two neighbouring blocks do not have an identical index, there still exists the probability of having identical ancestry (i.e. parents or grandparents, etc.). This feature can be used in bit saving, because there is no need to transmit the same information (similarity in family generation) for both of the neighbouring blocks.

6.3 Index Transmission or Storage

The methods proposed in Chapters 4 and 5 require two groups of information. A map that shows the boundaries of neighbouring blocks mapped to the same small codebook, and the index of the first member of each boundary that cannot be recovered by the indices of their neighbouring blocks. The new scheme for TSVQ like the proposed scheme in Chapters 4 and 5 requires transmitting two groups of information, a map that shows the genealogical relationship of the image block indices, and the bit required to show the genealogical differences. The map of genealogical relationship indicates the subtree to which the neighbouring blocks belong.

6.3.1 The Map of Genealogical Relationship

The map of genealogical relationships consists of information that shows whether two neighbouring blocks have an identical index, an identical parent, grandparents or any other identical ancestor. This map can be constructed by finding the neighbouring blocks with identical index, then those blocks that do not have an identical index with their neighbours but have an identical parent with their neighbouring blocks, and the procedure continues until arriving to the point where all the image blocks have the same patriarch; this point is the root of the tree.

6.3.2 The information of the genealogical differences

The map of genealogical relationship can produce the indices' information up to a stage that a block has relation with its neighbouring blocks, and the rest of the information should be transmitted. For example, let two blocks have the following indices respectively (read from left to right for root to node):

1 1 0 1 1

1 1 0 0 0

Up to three generations, the indices have identical ancestors indicated by 110, and after that all the remaining bits are required for reconstructing any of the two indices. In the case of a binary tree the first bit in the indices, where they are different, has to be transmitted for one of the blocks. If that bit is known for one of the blocks its value is automatically determined for the other. In the above example the difference starts from the fourth bit, and as that bit for the first index is a "1", it should be a "0" for the other block.

6.4 Rate of Index Compressed VQ method III

The required information rate for IC-VQ (III) consists of the information for the map of genealogical relationships and the genealogical differences. This section gives the required rate for each part.

6.4.1 The rate of genealogical information

In IC-VQ (III), the index of each block is first compared to the indices of two of its neighbouring blocks in the north- and west-side to find out to which of its neighbouring blocks it is most similar in ancestry. This information is 1 bpv. The rest of the genealogical relation bit rates are to show the information of the identical ancestry such as having identical

indices, parents, grandparents and so on. In the first step the required bit rate is 1 bpv, because among all the image blocks those which have an identical index with their north- or west-side neighbours have to be identified. The number of identified blocks with identical indices with their neighbours is $nP_{(W\cup N)_r}$, where n is the total number of image blocks, r is the TSVQ rate (or the number of tree layers), and $P_{(W\cup N)_r}$ is the probability that a given block has an identically indexed north- or west-side neighbour block in a TSVQ with an average rate of r bpv.

IC-VQ (III) identifies the blocks which have an identical parent with their neighbours in the second step. These blocks are among those that do not have an identical index with their neighbours. The blocks which remained from the first step (blocks without neighbours with an identical index) are $n\overline{P}_{(W\cup N)_r}$, where $\overline{P}_{(W\cup N)_r}$ is the probability of having neighbours in the north or west-side with a different index in an r bpv TSVQ. The rate to represent blocks with identical parents as found in the second step is $\overline{P}_{(W\cup N)_r}$. The process is continued in this fashion until the root of the tree is reached. Hence, the formula for the rate of genealogical information can be written as:

$$R_{gi} = 2 + \sum_{i=1}^{r} \overline{P}_{(W\cup N)_i} \quad \text{bpv} \tag{6.1}$$

where R_{gi} is the genealogical information rate, and $P_{(W\cup N)_i}$ is the probability of having blocks with identical indices for an i bpv TSVQ.

6.4.2 The rate for the genealogical differences

In TSVQ, if two neighbouring blocks have an identical index or parent, there is no need to transmit any side information for reconstructing the indices as explained in section 6.3.2. Based on this argument if two neighbouring blocks have k bits different from the tree node that their differences start, *k-1*

bits are required to reconstruct the index of any of them. This method results in the following formula for the rate of genealogical differences:

$$R_{gd} = \sum_{i=1}^{r} \left(P_{(W \cup N)_{(i-1)}} - P_{(W \cup N)_i} \right)(r-i) \text{ bpv} \qquad (6.2)$$

Where R_{gd} is the rate of genealogical difference, r is the average rate of the tree, and i is the subtree rate (the rate required to represent the node of the subtree to which the indices of the two neighbouring blocks belong). The proof for (6.2) is as follows. Suppose that the probability of having neighbours with identical indices, parents, grandparents are $P_{(W \cup N)_r}$, $P_{(W \cup N)_{(r-1)}}$, $P_{(W \cup N)_{(r-2)}}$. There is no need to send any bit for the case of blocks with identical indices or parents. The bit required is $(P_{(W \cup N)_{(r-2)}} - P_{(W \cup N)_{(r-1)}})((r\text{-}1)\text{-}r)$ for the case of neighbouring blocks with identical grandparents, because the information of $(P_{(W \cup N)_{(r-1)}} \; x \; 100)\%$ of the blocks has been previously transmitted, and $P_{(W \cup N)_{(r-2)}}$ is the fraction of blocks that have identical index, parents and grandparents all together. The pure values of those which have only an identical parent can be found by the difference, $(P_{(W \cup N)_{(r-2)}} - P_{(W \cup N)_{(r-1)}})$, and $((r\text{-})1\text{-}r)$ bits are required for index reconstruction in the case when two blocks have an identical grandparent. If this method is continued for the other cases and the bits for all the cases are added then (6.2) is obtained. The final rate for IC-VQ (III) is as follows:

$$R = 2 + \sum_{i=1}^{r} \left[\left(P_{(W \cup N)_{(i-1)}} - P_{(W \cup N)_i} \right)(r-i) + \overline{P}_{(W \cup N)_i} \right] \text{ bpv} \qquad (6.3)$$

IC-VQ (III) considers two neighbouring blocks in the compression process. One can consider three or more neighbours. However this may not necessarily improve the performance of IC-VQ (III), because the probability of identical ancestry for the case of two or more has little differences (about 4 to 8 %), and in the case of more than two there is need to transmit or store extra overhead information to represent the local genealogical relationship.

6.5 Methods to Improve the Rate of IC-VQ (III)

The rate formula of IC-VQ (III) indicates that the minimum value for the required rate is 2 bpv, and this is the overhead incurred in finding and representing the most similar neighbouring block based on the indices and the blocks having neighbours with identical indices. This rate can be reduced by using similar methods such as run-length coding as described in Chapters 4 and 5. The reason for the applicability of run-length coding is the possibility of having runs of similar symbols that convey the same information. As this part was extensively described in Chapter 4, it is not repeated here.

6.6 Index Compressed VQ method III based on FSVQ

The indices of images obtained by FSVQ do not have the genealogical information as described in TSVQ. The application of the new method in the case of FSVQ requires a method of giving the characteristics similar to the indices obtained from TSVQ to those obtained from FSVQ. In other words there is a need to give a tree shape to a FSVQ codebook, but still use the FSVQ scheme to quantise the images.

In TSVQ, two children of a tree node are those which have the most similarity based on the criterion, normally MSE, used for codebook generation. This characteristic generates indices for codevectors of a TSVQ in a way that they contain the genealogy information as previously explained. This characteristic can be virtually obtained for a FSVQ.

A simple and computationally efficient method, albeit not optimum as a tree-structured VQ codebook in terms of MSE, is to generate the virtual tree on the basis of the similarity between the energy of codevectors and consider every pair of codevectors which have the most similar energy as the children of a tree

node. Figure 6.3 to 6.5 demonstrate the procedure of this scheme. The codevectors are ordered based on their energies. This process is shown in Figure 6.3. Figure 6.4 shows the pairs of codevectors that construct the codevector of the node in the next layer of the virtual tree, and Figure 6.5 illustrates how to continued and have a complete virtual tree.

Figure 6.3: The ordered codevectors based on their energies

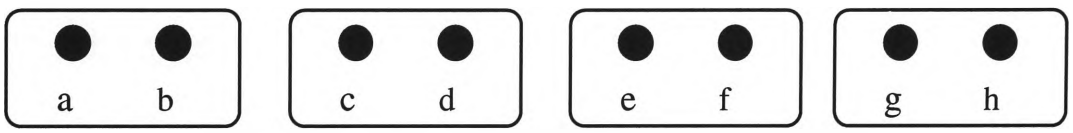

Figure 6.4: Each pair of codevectors are merged

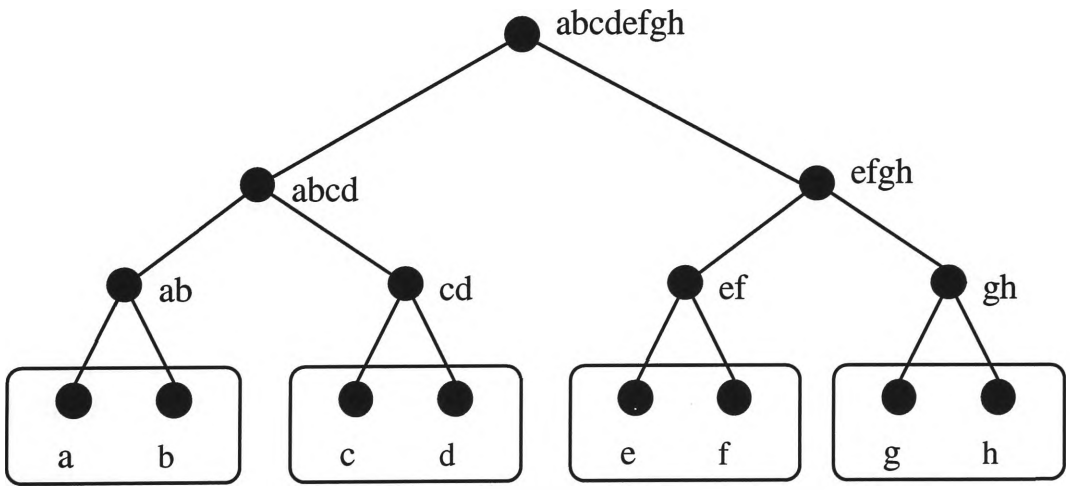

Figure 6.5: A virtual tree construction

6.7 Simulation Results and Discussion

This section presents the simulation results and discussion of the new scheme for TSVQ and FSVQ based on virtual tree structure. The results are based on 512x512 size image and block size of 4x4. The effects of block size for the case of index compression by IC-VQ (III) are discussed as well.

6.7.1 Simulation results and discussion for TSVQ

The result for IC-VQ (III) has been compared with IC-VQ (II) based on TSVQ, JPEG and predictive pruned TSVQ (PPTSVQ), FS-VQ and address VQ. Two images are coded by IC-VQ (III) and JPEG, to evaluate the subjective quality. The results obtained are based on the test image, *Lena* (512x512 pixels) using 4x4 pixels block size. The PSNR has been calculated using (4.16).

6.7.1.1 Comparison between IC-VQ (III) and IC-VQ (II)

Figure 6.6 presents the results of IC-VQ (II) (graph labelled "A"), and IC-VQ (III) (graph labelled "B") based on TSVQ. It can be seen that IC-VQ (III) has a better performance than IC-VQ (II) in all ranges of bit rate. The differences at low bit rates is low, because the codebook size for the both codecs is small and results in little differences between the low and high activity areas of the image, therefore the variable rate bit assignment procedure is less effective. At higher rates the difference is about 0.7 dB, and at the same PSNR (30.5 dB) the difference in rate is about 0.07 bpp. The reason for the difference at higher rates is that the image blocks have more choices to find the best match from the codebook, consequently the possibility of mapping the neighbouring blocks onto a small and fixed size subset of the original codebook decreases, and IC-VQ (III) alleviates this problem by a variable mapping the neighbouring blocks.

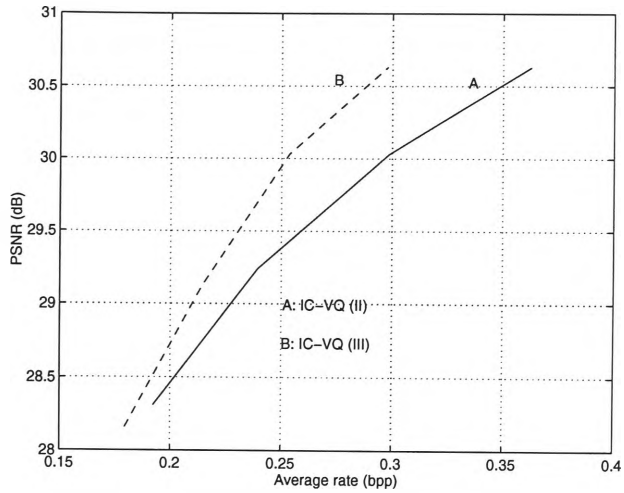

Figure 6.6: Results of IC-VQ (II) and (III) based on TSVQ

6.7.1.2 Comparison between IC-VQ (III) and predictive VQ

Figure 6.7 presents the results of PPTSVQ (graph labelled "A") [Lookabaugh 1993] and IC-VQ (III) (graph labelled "B"). The results indicate that IC-VQ (III) is able to produce better performance at rates less than 0.3 bpp. Between 0.1 and 0.15 bpp the difference varies from 4 to 1 dB.

The reason for the differences between PPTSVQ and IC-VQ (III) at very low bit rates is that the predictive coder relies on prediction and a small codebook size to represent all the image blocks with any activity. This approach has two problems. Firstly, prediction does not perform well on high activity areas such as edges. Secondly, a small codebook cannot represent a wide variety of block shapes. In the case of IC-VQ (III), at very low bit rates, the most performance improvement derives from exploiting the inter-block correlation, rather than a small codebook. The variable bit allocation of IC-VQ (III) is image adaptive while it is not the case for PPTSVQ.

IC-VQ (III), in terms of codebook design is much simpler than the predictive coder because it is based on the basic TSVQ. The design process of

PPTSVQ in its simplest form involves calculating the prediction coefficients, generating the residual of vectors between the predicted values and the original vector, generating the tree-shaped codebook for the residual vectors and then pruning the tree. The method of optimising the codebook of the PPTSVQ is complex and requires several iterations [Chang 1986, Lookabaugh 1993, Gersho 1992].

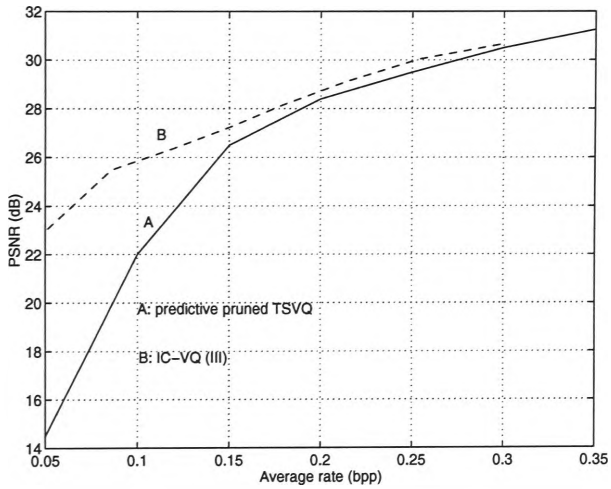

Figure 6.7: Results of IC-VQ (III) based on TSVQ, and PPTSVQ

In terms of simplicity of encoding, both encoders are using tree-structured codebooks which have the same block quantisation time; the difference is that in the predictive coder, the residual has to be obtained, and in IC-VQ (III) a comparison between the index of each block with the indices of two of its neighbours has to be performed plus Huffman coding of the genealogical map. As a conclusion, the encoding procedures of these two schemes do not have significant differences, but in terms of design procedure and coding performance at bit rates less than 0.3 bpp, IC-VQ (III) is preferable.

6.7.1.3 Comparison between IC-VQ (III) and JPEG

Figure 6.8 demonstrates the results of JPEG and IC-VQ (III) based on TSVQ. IC-VQ (III) outperforms JPEG at rates less than 0.21 bpp. At lower

for the poor performance of JPEG at low bit rates was stated earlier in Chapter 5 Section 5.5.

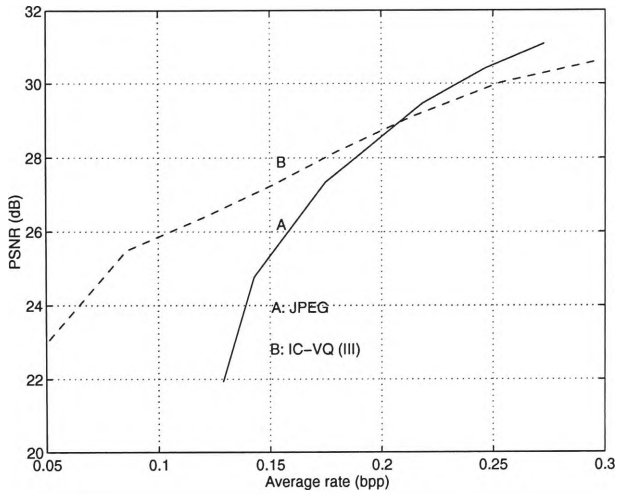

Figure 6.8: Results of IC-VQ (III) based on TSVQ, and JPEG

Two test images, *Lena* and *Couple*, were chosen to evaluate the subjective quality. These two images are coded at about the same rate by IC-VQ (III) and JPEG. The results for *Lena* are illustrated in Figures 6.9 and 6.10, and the results for *Couple* are shown in Figures 6.11 and 6.12. The results obtained by JPEG contain noticeable blocking, while this is not the case for IC-VQ (III). The reason for the blocking effect associated with JPEG is that this scheme assigns more bits to active areas. This approach has problems at low bit rates; there is not enough bits to represent low activity areas, and this results in discontinuities at the blocks boundaries.

6.7.1.4 Comparison among IC-VQ (III), FS-VQ and Address VQ

The coding performance of IC-VQ (III) based on TSVQ is about the same with FS-VQ, but with considerably less computation than FS-VQ whether in design or search process. Address VQ shows about 0.5 dB better results than IC-VQ (III). However the problems associated with address VQ in terms of

codebook generation, memory requirement and search complexity make IC-VQ (III) preferable for application.

6.7.2 Simulation results and discussion for FSVQ

The result for IC-VQ (III) based on FSVQ has been compared with IC-VQ (II), predictive FSVQ, and JPEG. The performance of IC-VQ (III) based on TSVQ and FSVQ, at the rate that FS-VQ and address VQ are compared with IC-VQ (III) based on TSVQ, is the same. Hence, there is no need to compare the performance of IC-VQ (III) based on FSVQ with these two coders. Two coded images, by IC-VQ (III) based on FSVQ and JPEG, are presented in Figures 6.16 to 6.19 to evaluate the subjective quality.

6.7.2.1 Comparison between IC-VQ (III) and IC-VQ (II)

Figure 6.13 presents the results of IC-VQ (II) and (III) based on FSVQ. It can be seen that the performance difference between methods II and III is about 0.6 dB at bit rates above 0.2 bpp and these two schemes have about the same performance at very low bit rates. The reason was stated earlier for the case of IC-VQ (III) based on TSVQ. Both of these schemes have the same problem when the channel noise is considered; noise propagation.

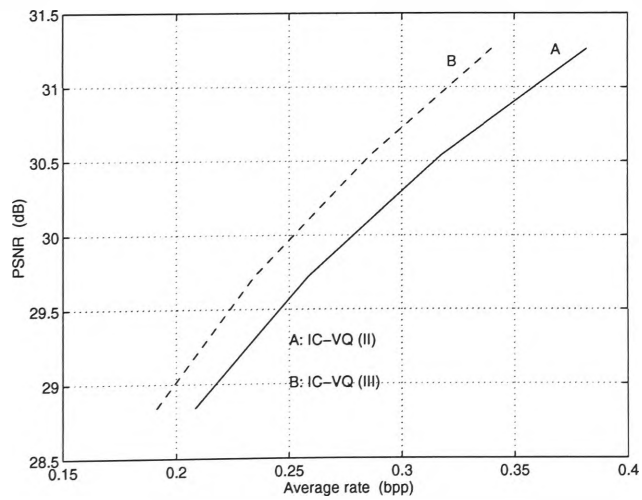

Figure 6.13: Results of IC-VQ (II), and (III) based on FSVQ

6.7.2.2 Comparison between IC-VQ (III) and predictive VQ

Figure 6.14 presents the results of predictive FSVQ [Lookabaugh 1993], and IC-VQ (III). IC-VQ (III) shows more than 2 dB better performance on average. At very low bit rates the coding performance of IC-VQ (III) is about 4 dB better than predictive FSVQ. In terms of implementation, the design process of the predictive coder [Chang 1986, Lookabaugh 1993, Gersho 1992] is more complex than IC-VQ (III), which requires only a simple codebook such as a memoryless FSVQ. The reason for the differences, as stated before, are the inability of prediction on high activity areas and the use of a small codebook for the representation of all the residuals in predictive VQ coding at very low bit rates.

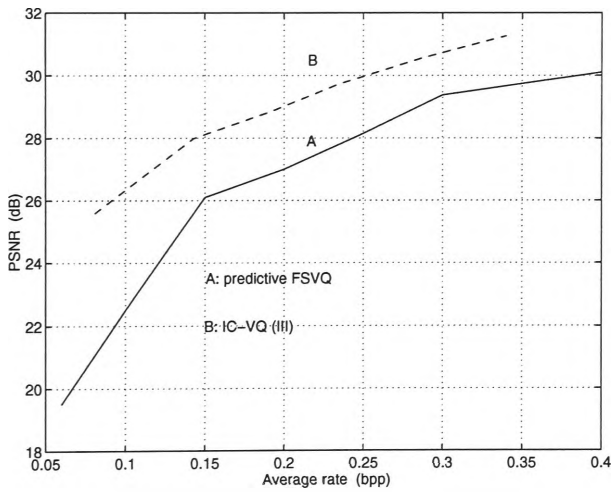

Figure 6.14: Results of IC-VQ (III) based on FSVQ and predictive FSVQ

6.7.2.3 Comparison between IC-VQ (III) and JPEG

Figure 6.15 shows that IC-VQ (III) outperforms JPEG at bit rates less than 0.22 bpp. This rate threshold is image dependent. For example for a more active image such as *Couple* this rate is about 0.3 bpp. This is because JPEG requires more bits to represent the high activity blocks without considering the inter-block correlation, while there exist some correlation in high activity areas which is easily exploited by IC-VQ (III).

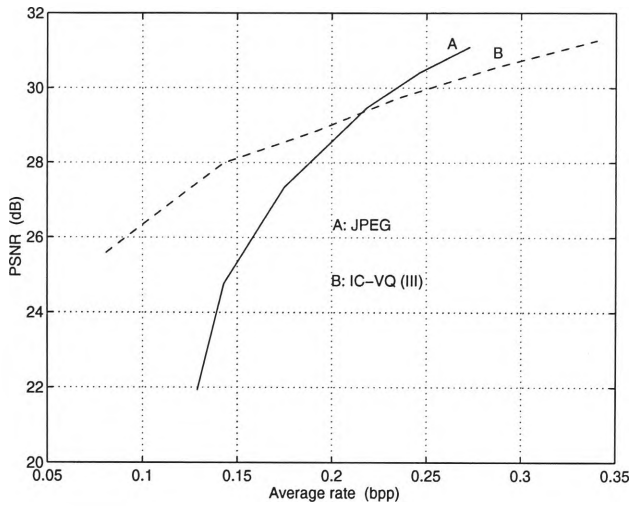

Figure 6.15: Results of IC-VQ (III) based on FSVQ and JPEG

Two test images as in the previous section were selected to evaluate the subjective quality. The coded versions of these two images with about the same rate were obtained using IC-VQ (III) and JPEG. They are respectively shown in Figures 6.16 to 6.19. The quality of coded images using VQ are visually much better than JPEG. While the coded images have higher rates than those in section 6.7.2, the JPEG codec still introduces noticeable blocking effect in smooth areas such as face which is visually important.

6.7.3 The effect of block size and image activity on the performance of IC-VQ (III)

Chapter 3 showed that small block size vector quantisation increases the probability of having more identically indexed neighbouring blocks and consequently more neighbouring blocks with indices having identical ancestors. Thus the coding performance of IC-VQ (III) over its counterpart VQ can be improved if small sized blocks are used. The other characteristic of images which affects the performance of IC-VQ (III) is the image activity; highly active images have less identically indexed neighbouring blocks, or blocks with indices having identical ancestor, and consequently IC-VQ (III)

has less performance on these types of images in comparison with images with low to moderate activity. Similar arguments can be advanced for IC-VQ (I) and (II).

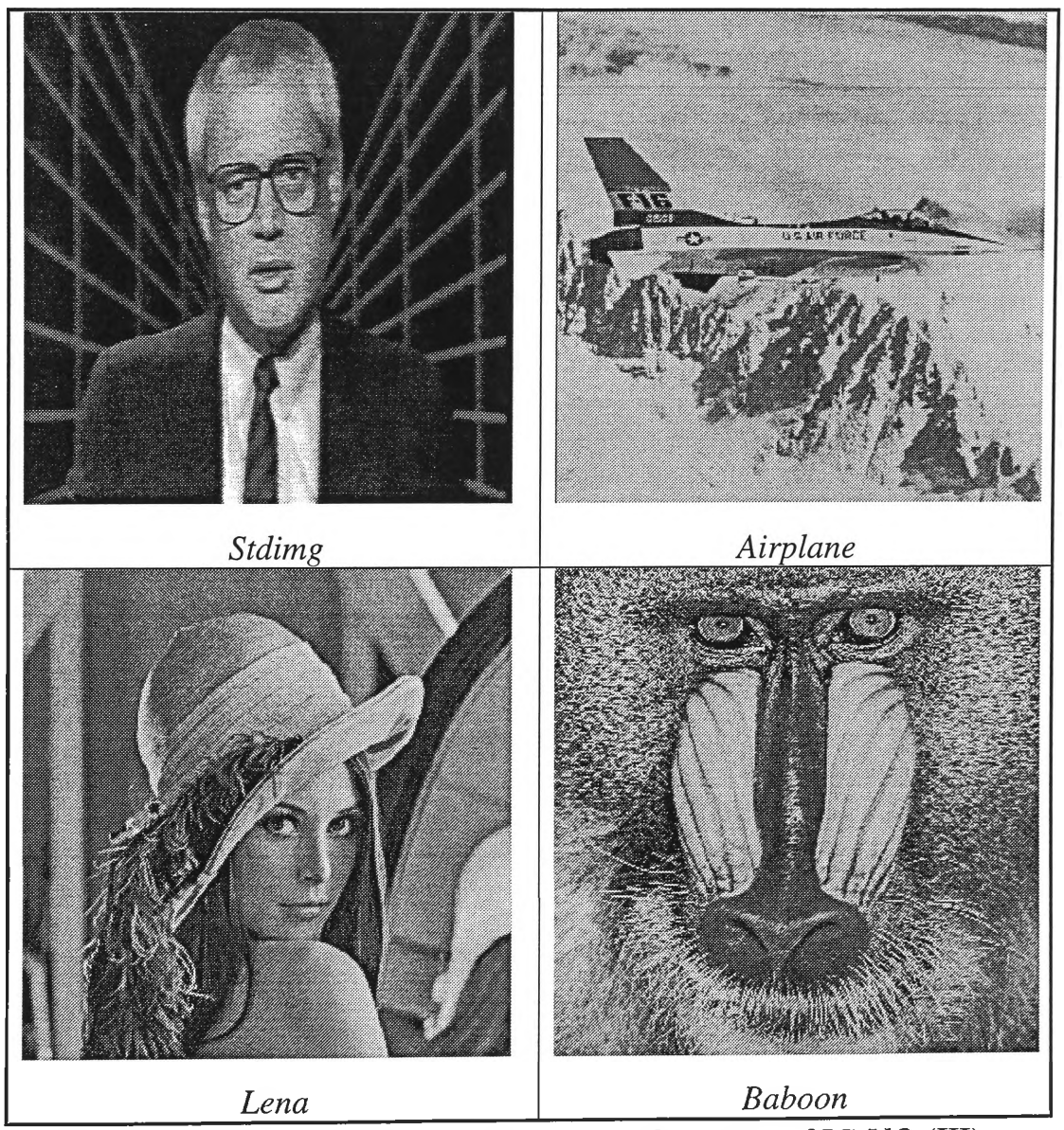

Figure 6.20: Four images for testing the performance of IC-VQ (III)

Four images (Figure 6.20) with varying activities have been selected to illustrate the above arguments. The complete results for each image are presented in appendix B, and a summary of results based on an approximate difference between IC-VQ (III) and FSVQ is shown in Figure 6.21. First, it

can be seen that IC-VQ (III) based on smaller block size gives better improvement over its counterpart VQ coder with the same block size, and secondly the performance of IC-VQ (III) highly depends on the image activity. The coding performance of IC-VQ (III) for less active images is better than highly active images.

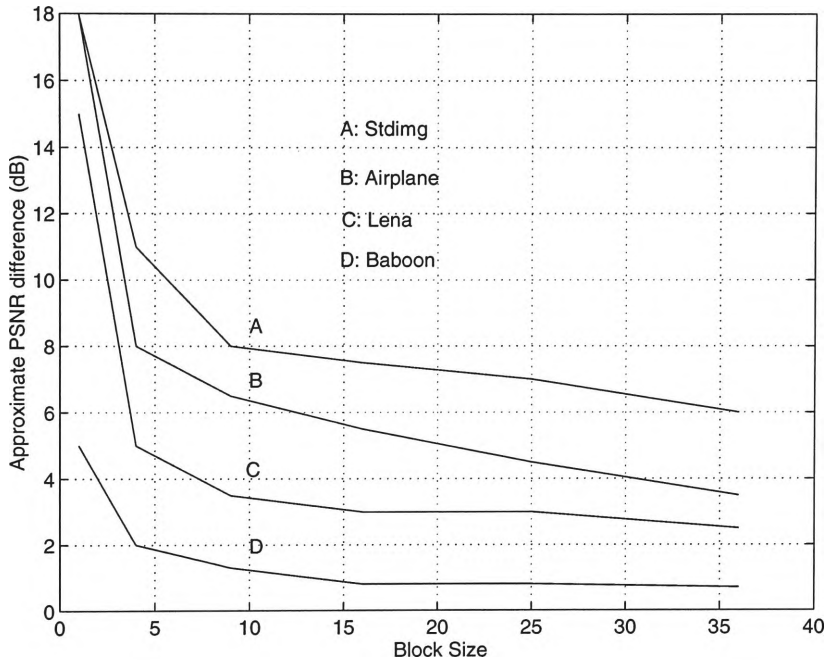

Figure 6.21: The effect of block size and image activity on the performance of IC-VQ (III)

6.8 Summary

A novel method of image coding, IC-VQ (III), capable of providing significant results at low bit rate is proposed. This method compresses the indices of quantised images based on the fact that neighbouring blocks, in natural images, are highly correlated, and this correlation exhibits itself among the indices of neighbouring blocks in a way that neighbouring blocks are mapped onto a small subset of the original VQ codebook; the size of the small codebook depends on the neighbouring blocks' activity. The

genealogical relationship among the indices has been used to exploit this characteristic.

The performance of this scheme in terms of PSNR versus average rate was compared with IC-VQ (II), some predictive VQ coders, FS-VQ, address VQ and JPEG. The results show that this scheme has better compression capability in terms of objective quality over most of these schemes at bit rates less than 0.3 bpp, depending on the activity of the coded image, and gives a comparable result with address VQ with much less complexity. The subjective quality of this scheme has been compared with JPEG. This scheme, at low bit rates, introduces less noticeable distortion when compared with JPEG.

Figure 6.9: *Lena* at 0.2536 bpp by IC-VQ (III) based on TSVQ (with PSNR 30.03)

Figure 6.10: *Lena* at 0.2463 bpp by JPEG (with PSNR 30.41)

Figure 6.11: *Couple* at 0.2647 bpp by IC-VQ (III) based on TSVQ (with PSNR 27.40)

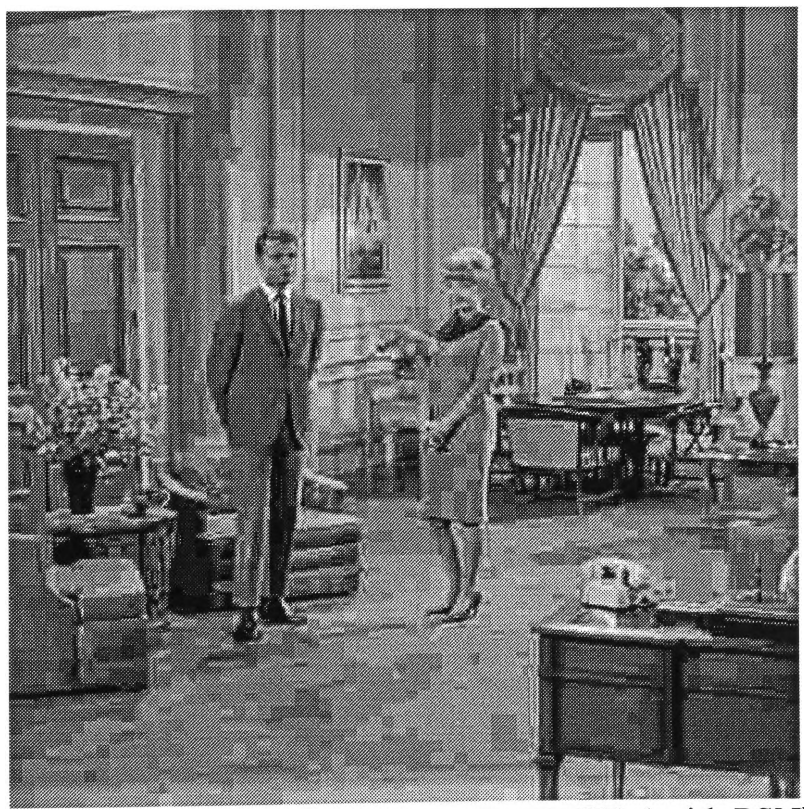

Figure 6.12: *Couple* at 0.2686 bpp by IC-VQ (III) (with PSNR 27.24)

Figure 6.16: *Lena* at 0.2861 bpp by IC-VQ (III) based on FSVQ (with PSNR 30.54)

Figure 6.17: *Lena* at 0.2728 bpp by JPEG (with PSNR 31.09)

Figure 6.18: *Couple* at 0.2942 bpp by IC-VQ (III) based on FSVQ (with PSNR 28.02)

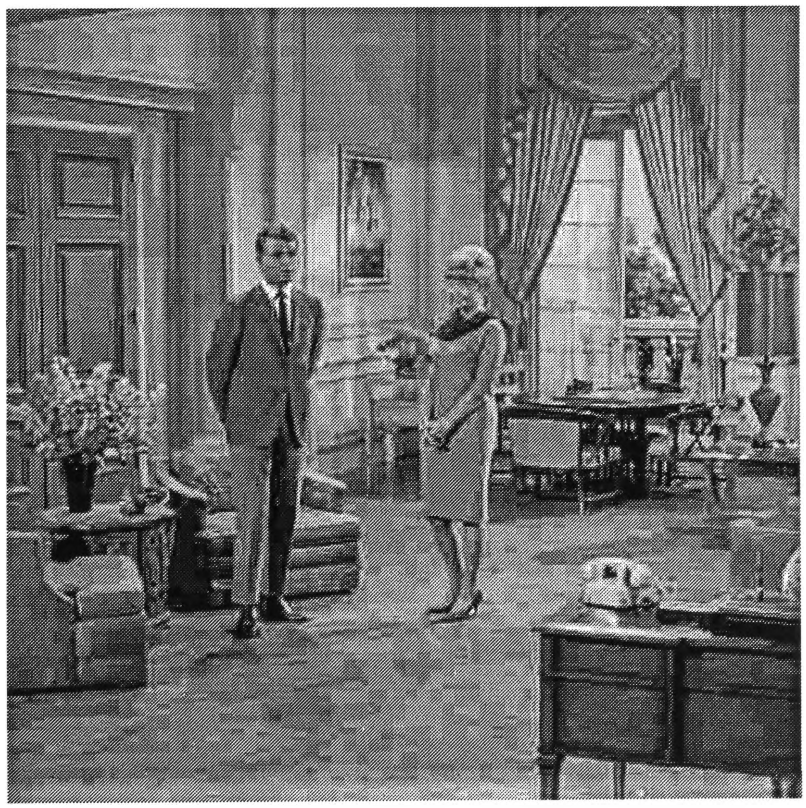

Figure 6.19: *Couple* at 0.3116 bpp by JPEG (with PSNR 28.06)

Chapter 7

Index Compressed Image Adaptive Vector Quantisation

7.1 Introduction

A class of VQ schemes to improve the subjective quality of coded images at low bit rates is image-based VQ. VQ schemes can be partitioned into two groups depending on how the codebook is generated. When several images are used as the training set in generating the codebook, it is called universal codebook based VQ, and when the image to be encoded is the only training set used in generating the codebook, it is referred to as an image-based codebook. Encoders using an image-based codebook are sometimes referred to as image adaptive VQ (IAVQ). These coders result in lower distortion than a similarly sized universal codebook based VQ, and can better represent the active areas such as edges [Goldberg 1986], because they follow the characteristics of the image blocks more closely.

The basic image adaptive VQ (IAVQ) proposed by Goldberg et al [Goldberg 1986] has three disadvantages. Firstly, it bears a great computational burden of codebook generation because it is based on the LBG algorithm of Linde et al. [Linde 1980]. Secondly it requires transmitting the overhead information (the generated codebook), which may result in having an IAVQ coder with higher distortion for the same rate when compared to a universal VQ coder. Finally, it neglects the inter-block correlation.

The first problem can be solved by employing a fast clustering algorithm, and a number of these have been proposed in the literature. There exist fast clustering schemes such as real-time codebook re-transmission [Huang 1994], fast pairwise nearest neighbourhood [Equitz 1989], and Kohonen's

self organising feature map [Nasrabadi 1988]. Beside the fast clustering schemes the development of modular VLSI architectures for real-time full-search based vector quantisation [Park 1993] is another step in this regard. Another method of solving the problem associated with the clustering algorithm is developed in [Panchanathan 1991]; a sub-optimum codebook is generated without using a clustering algorithm. Chen and Sheu [Chen 1994] introduced the Gold-Washing method which is able to generate the codebook on the fly.

The problem associated with overhead bit rate becomes important when the required rate for an IAVQ results in having a coder with the same performance as a universal codebook based VQ. The general approach to this problem is the use of some lossy coding scheme for codevectors transmission. This idea is motivated by the existence of intra-vector (or intra-block) correlation among the pixels of the codevectors. So far, three lossy schemes have been reported; variable length transform coding [Wang 1992], JPEG [Huang 1994], and universal codebook based vector quantisation with a large codebook [Zeger 1992, 1994]. In situations where the overhead rate is low, applying a lossy scheme does not result in appreciable improvement.

The coding performance of IAVQ can be improved by considering the inter-block correlation. Wang et al. [Wang 1992] considered this approach by combining variable-length transform coding and image adaptive vector quantisation. The method essentially uses an image based codebook with large image blocks in transform domain. It is worth noting that the size of the blocks in Wang scheme is larger than that used by Goldberg et al. [Goldberg 1986]. The larger block size together with the transformation

employed in the encoder make the proposed scheme more complex than the basic image adaptive VQ. Furthermore, the coding performance of the codec proposed by Wang et al. is poor at low bit rates (less than 0.3 bpp) [Wang 1992, page 899].

This chapter exploits the inter block correlation through an index compression scheme. Subsequently this scheme is called index compressed IAVQ (IC-IAVQ). The rest of the chapter is organised as follows: Basic IAVQ, and its combination with index compression is explained in section 7.2 and 7.3, and the related results, comparison with other schemes and discussion are presented in section 7.4.

7.2 Image Adaptive Vector Quantisation

In a basic IAVQ scheme, the image to be encoded is partitioned into identically sized blocks of pixels and a codebook is generated using these image blocks. For each image, the generated codebook is transmitted along with the image's block indices to the receiver. The generated codebook can be transmitted via a lossless or a lossy scheme. The rate, r, of the basic IAVQ can be obtained from the following expression:

$$r = \frac{\log_2 N}{k} + \frac{RN}{M} \quad \text{bpp} \tag{7.1}$$

where, N is the codebook size; k is the block size; M is the total number of identically sized blocks into which the image is partitioned; and R is the number of bits allocated to each component of the codevector in the codebook. The first term of (7.1) is the contribution to the rate from indices representation. The second term, which is the overhead rate, arises from the

codebook size. A reduction of codebook size will definitely reduce the contribution of this term to the overall rate. However, a reduction in codebook size leads to an increased distortion in the reconstructed image.

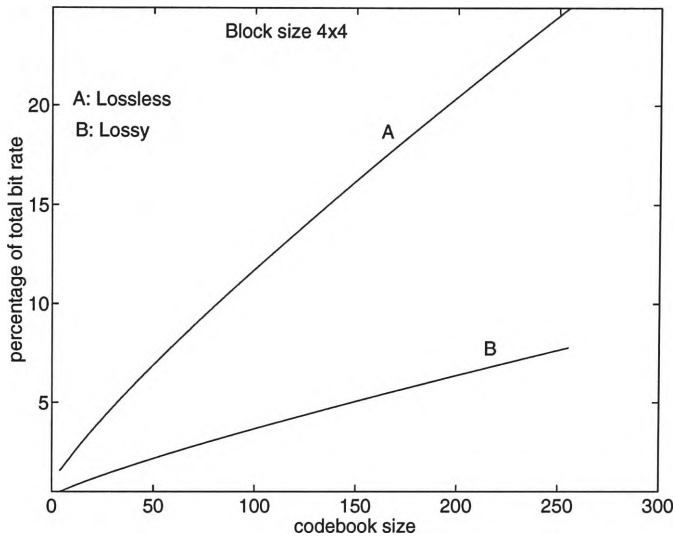

Figure 7.1: Bit rate increase using lossless (graph labelled "A") or lossy (graph labelled "B") coding of the codevectors (4x4 block size)

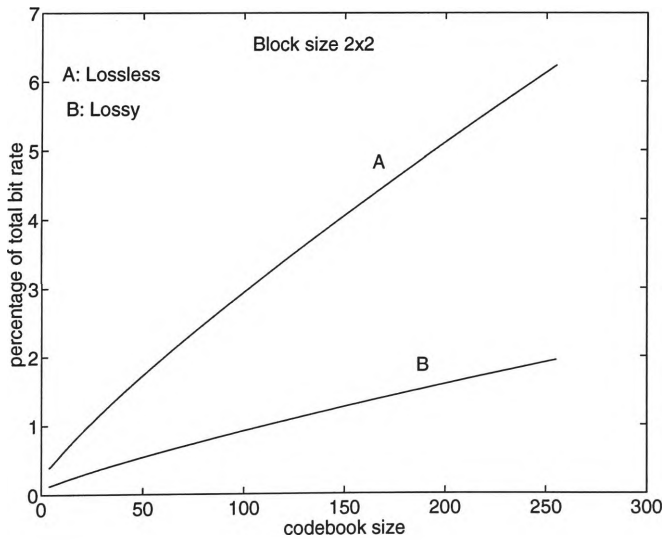

Figure 7.2. Bit rate increase using lossless (graph labelled "A") or lossy (graph labelled "B") coding of the codevectors (2x2 block size)

Apart from reducing the codebook size, other researchers [Wang 1992, Huang 1994, Zeger 1992, 1994] have employed a lossy coding scheme to reduce the overhead rate. Figures 7.1 and 7.2 show a comparison of the contribution of the overhead rate (percentage) to the total bit rate when a lossy (2.5 bits/pixel) and lossless (8 bits/pixel) coding schemes are used. The results are for the case of a *512x512* pixels image with block sizes of *4x4* and *2x2* pixels respectively. The percentage of overhead rate is calculated using the following formula:

$$percentage = \left(\frac{\frac{RN}{M}}{\frac{\log_2 N}{k}} + \frac{RN}{M} \right) x100 \approx R\left(\frac{kN}{M \log_2 N} \right) x100 \qquad (7.2)$$

It can be seen that when the vector size is less than or equal to 16 and the codebook size is less than 128, the contribution of overhead rate is less than 10 percent of the total rate, and there is little difference between applying a lossy or a lossless coding scheme. However, for a bigger codebook the difference is significant and it is worthwhile considering the use of a lossy coding scheme in compensating the side effect of overhead rate.

7.3 Index Compressed Image Adaptive Vector Quantisation

The block diagram of IC-IAVQ is shown in Figure 7.3. In the encoder, the input image is partitioned into *nxn* pixels blocks, which are used to generate an image-based codebook via a clustering algorithm such as the LBG algorithm. The codevectors are encoded and transmitted (stored) using a lossless or lossy scheme. The indices, after compression by one of the lossless coding schemes proposed in chapters 4 to 6, are also transmitted or stored. At the decoder a lookup table (codebook) based on the decoded codevectors is constructed; in addition, the compressed indices are decoded.

The vector quantised version of the original image is reconstructed by using the decoded indices and the codebook.

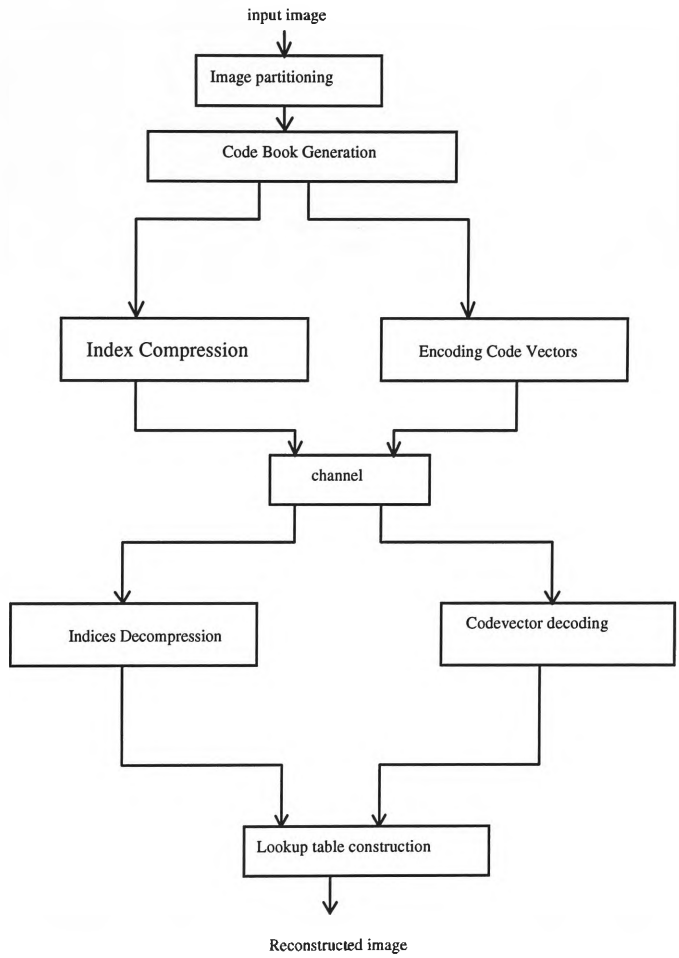

Figure 7.3. Block diagram of an IC-IAVQ codec

7.4 Simulation results and discussion

The proposed scheme has been tested on three images with low, moderate and high activity areas; *Airplane, Lena*, and *Baboon* (Figure 7.4). The results of the new scheme in terms of objective quality has been compared with the basic IAVQ (for a block size of 4x4 pixels) and JPEG coding standard. The generated image adaptive codebooks were transmitted by a

codebook containing vectors of size of 2x2 pixels, with 1024 codevectors (2.5 bpp). The PSNR has been calculated using (4.19).

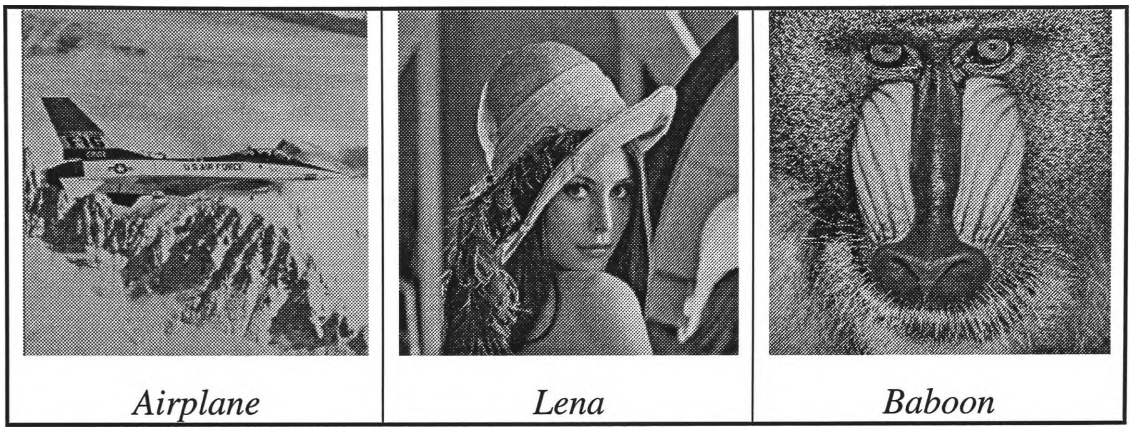

Figure 7.4: Test images for IC-IAVQ

Figures 7.5 to 7.7 show the results of simulation carried out on the 3 images for the case where the image-based VQ coder is full search. All the figures show that the combination of image adaptivity and index compression (graphs labelled "B") results in a significant improvement, up to 4 dB depending on the image activity, when compared with image adaptivity alone (graphs labelled "A"). The improvement is more significant for images with low to moderate activity, because for those images IC-VQ is more effective.

The coding performance of IC-IAVQ shows superiority over JPEG coding standard at bit rates less than 0.25 bpp. In case of images with high activity areas, IC-IAVQ outperforms JPEG over a wide range of bit rates. The performance of IC-IAVQ is about 1 dB better than JPEG (Figure 7.7), and this performance cannot been achieved by universal based VQ coders in combination with index compression. The reasons for this result are the application of image-based codebook for images with high active areas and the exploitation of inter-block correlation by index compression. Index

compression performs in such a way to exploit the inter-block correlation even in highly active areas, while JPEG ignores this correlation. The universal based codebook cannot represent areas with any form of activity, while this is possible for the image-based codebook.

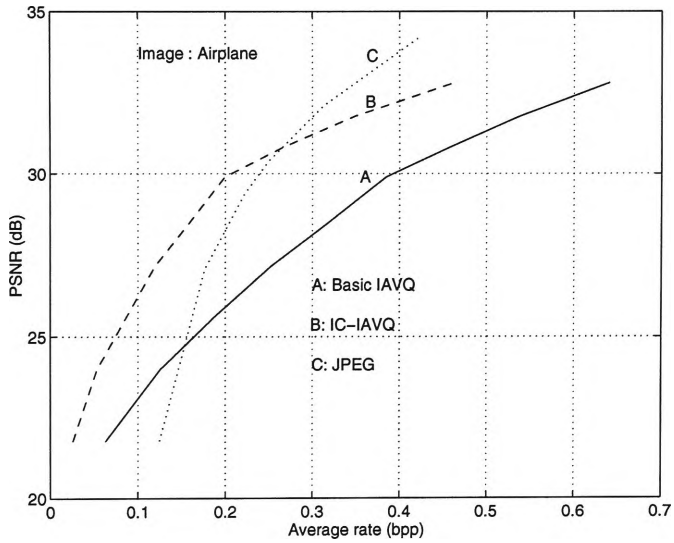

Figure 7.5: Results of IC-IAVQ with *Airplane* as test image

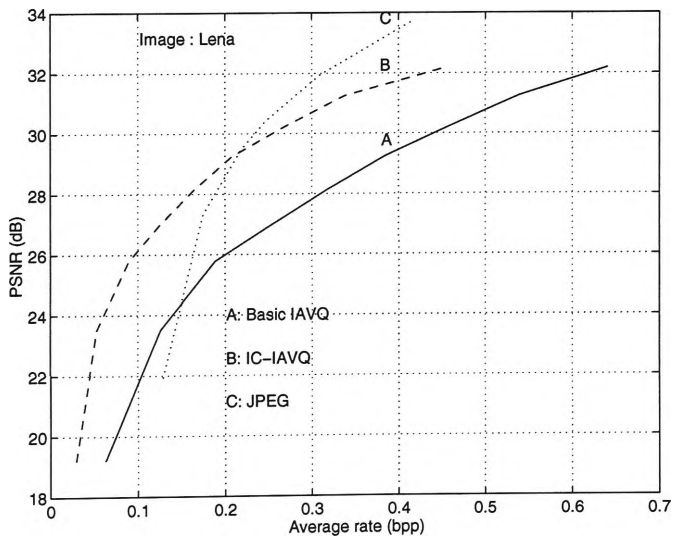

Figure 7.6: Results of IC-IAVQ with *Lena* as test image

7.5 Comparison between Wang et al. Scheme and IC-IAVQ

IC-IAVQ has two important advantages over the scheme proposed by Wang et al. [Wang 1992]. Firstly, the encoder is much simpler, and secondly it provides a much better results at low bit rates. For example Wang scheme

gives 30.91 dB PSNR at 0.7238 while IC-IAVQ gives 30.8 dB PSNR at 0.3 bpp for the same image (Lena).

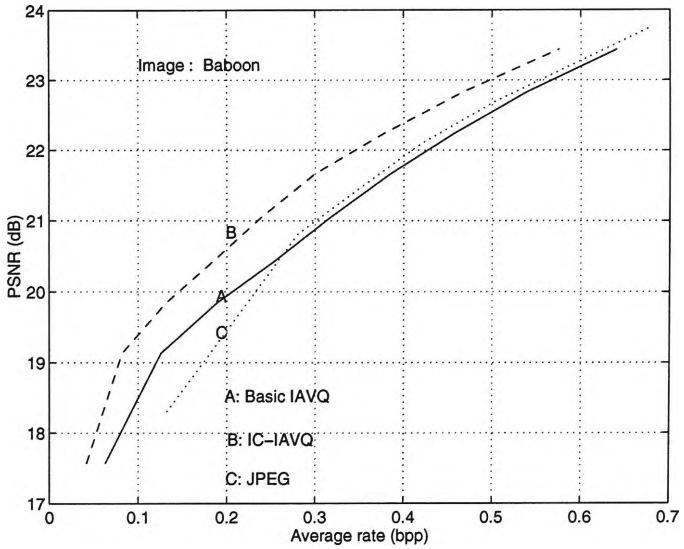

Figure 7.7: Results of IC-IAVQ with *Baboon* as test image

7.6 Summary

This chapter introduces an image adaptive VQ schemes in which the inter-block correlation has been exploited through an index compression algorithm. In IC-IAVQ method, the codebook is generated based on the image to be encoded, and the indices of the encoded image are compressed via an index compression algorithm. The aim of index compression is to exploit the interblock correlation which has been neglected in the basic IAVQ scheme. The results show that the new method results in significant, up to 4 dB, improvement over the basic image adaptive VQ. The application of IAVQ in combination with index compression obtains significant improvement over JPEG coding standard for images that contain areas with high activity. IC-IAVQ shows better performance with simpler encoder than Wang Scheme.

Chapter 8

Index Compressed Residual Adaptive Vector Quantisation

8.1 Introduction

The main goal of IAVQ, proposed by Goldberg et al. [Goldberg 1986], was to encode the image by a non-stationary codebook thus enabling a more subjectively pleasing reconstruction of active image areas. However, in IAVQ, generating an adaptive codebook for all the image blocks results in more accurate encoding of the low activity areas which may not necessarily result in significant subjective quality improvement of those areas over the universal codebook based VQ. Furthermore this method results in inefficient bit allocation from subjective quality points of view, because more bits are allocated to transmit low activity areas which can be universally encoded with barely noticeable distortion.

An alternative strategy to improve the performance of IAVQ is to use the universal codebook-based VQ with an index compression scheme for all the image blocks, and an adaptive codebook-based VQ to encode the residual of the active areas of the image. Using a universal codebook-based VQ, the low activity areas of the image can be reconstructed with low distortion and the

active areas can be approximated. The assignment of more bits to encode the residual of active areas in another step results in a fine reproduction of active areas.

If the residuals between the approximate version (the quantised image using the universal codebook-based VQ) and the original values of active blocks are further encoded, it is possible to reproduce the active areas fairly accurately. This method allows the use of a small codebook to encode the residual of active blocks. The codebook for residual vectors can be obtained universally or based on the image itself. The second method requires a smaller codebook, but more computation, and is suitable for images with areas with high activity. Whichever codebook is used, universal or image-based, this algorithm has a major computational advantage over the basic IAVQ. In the case of a universal codebook, the codebook which is needed for the residual vectors need not be generated for each image. In the case of an image based codebook, the training vectors and the size of the codebook for the residual vectors are small.

The adaptivity of this scheme derives from the use of the image characteristics to detect and appropriately allocate bits to the active regions of the image rather than the use of an image-based codebook for all the image blocks, since the codebook can easily be chosen to be universal.

Perhaps, from a classification point of view, this method can be considered as a multistage classified VQ [Juang 1982, Frost 1991, Barnes 1990, Ramamurti 1986]. A multistage classified VQ consists of several stages, and in each stage a quantised version of the residual of the previous section is obtained. In classified VQ (CVQ) the vectors are classified based on their activity into

several classes, and each block is vector quantised according to its class. A codebook with the capability of edge classification is used to encode the high activity blocks.

The new method consists of two stages. The first stage is an index compressed VQ and the second stage is a CVQ where there are two classes, the active areas and the rest of image. Only the residual of active blocks are encoded. It is pertinent to outline some of the salient differences between the new method and CVQ. The number of classes usually considered in CVQ is high with the consequent increase in rate introduced by the overhead of the class addresses. In the new method there is no overhead incurred in representing the two classes. Another important difference between the two methods lies in the allowable edge orientation of the active blocks allowable. CVQ places more restrictions on this important feature while the new scheme, especially in its image-based codebook form, allows any type of edges to be encoded.

The organisation of this chapter is as follows. The next section introduces the method in detail. Section 8.3 describes the method of partitioning the image into low and high active areas. Section 8.4 presents the simulation results and compares the new method with IC-IAVQ.

8.2 Index Compressed Residual Adaptive VQ

Figure 8.1 shows the block diagram of the system, Index Compressed Residual Adaptive VQ (IC-RAVQ), encoder. The system has to transmit two groups of information, the indices generated from encoding the image and the indices from encoding the residual. Therefore the system consists of three parts, a direct encoder for the image, an encoder for the residual of active

blocks and an interface to find out the active blocks and their residual. Next each section of the system is described.

The input image is vector quantised using a universal codebook, and the generated indices are compressed and transmitted. An index classifier identifies the active blocks and the address of their locations. The details of the index classifier is given in Section 8.3. The system then uses the address of the active blocks, a totally decoded version of the image and the original image to obtain the residual of the active blocks. Finally, the residual vectors are encoded by an adaptive codebook.

As aforementioned, the codebook used in encoding the residual vectors can either be universal or image-based. In the block diagram of Figure 8.1 a universal codebook has been used. If an image-based codebook is used, the system requires two additional sub-systems: codebook generation block and codebook encoding.

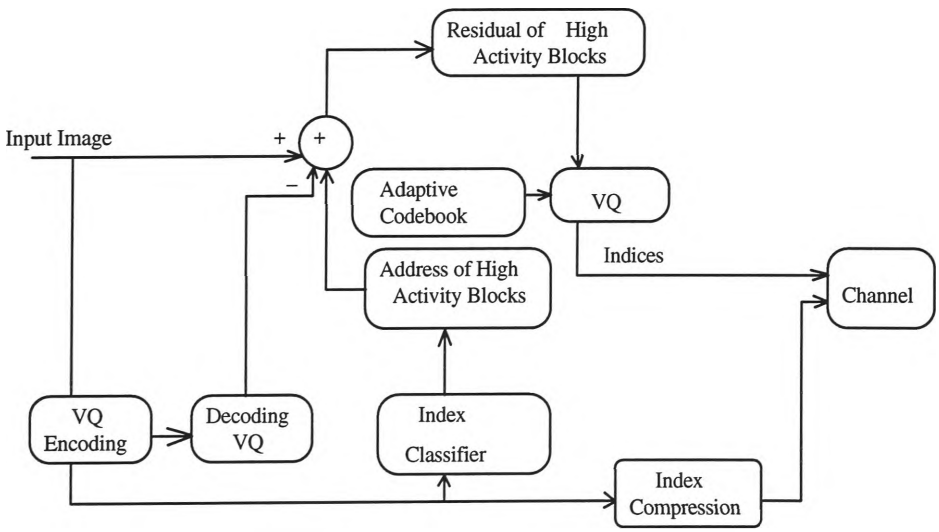

Figure 8.1: Encoder of IC-RAVQ

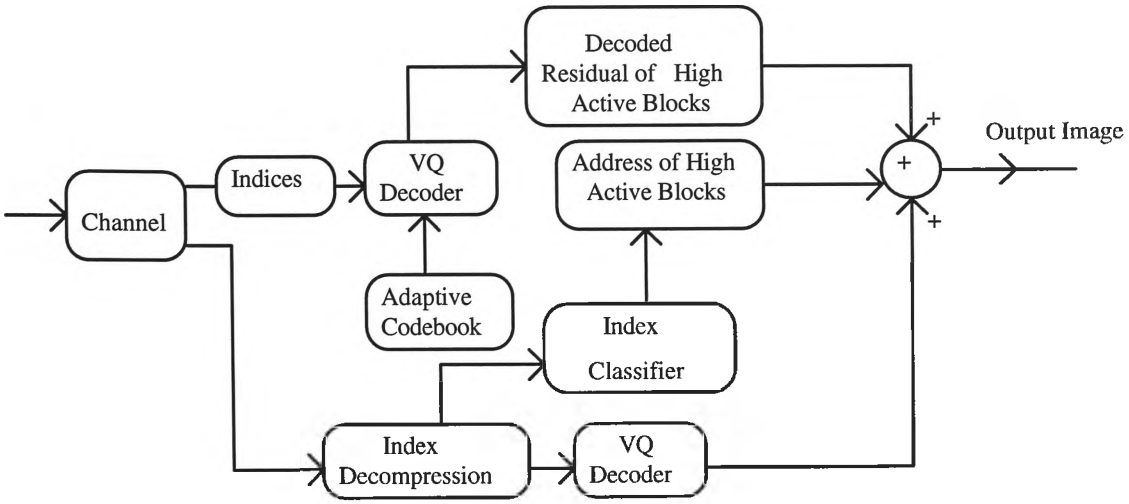

Figure 8.2: Decoder block diagram of IC-RAVQ

The corresponding decoder is shown in Figure 8.2. Two groups of information need to be decoded: the universally encoded image and the residual vectors. The situation in which an image-based codebook has been used requires a decoder to reconstruct the codebook. As shown in Figure 8.2, the indices resulting from directly encoding the image are recovered through a decompression algorithm, and a decoded version of the image is constructed. The decompressed indices also allows the identification of the addresses of active blocks and hence a recovery of the residual vectors and the quantised version of the image can be used to reconstruct the encoded image.

8.3 Image Partitioning into High and Low Activity Blocks

In this section a method of classifying the image blocks into high and low activity categories is described. This method has been tested on several images and the results are in agreement with a subjectively good segmentation. Apart from the simplicity and the speed of the classification method, it can be used in the quantisation domain, thus making it unnecessary to transmit any overhead information to the decoder in order to identify the active blocks.

Low activity blocks are usually characterised by a low variance about the mean pixel value. Hence, a block can be easily classified into the low activity category if it satisfies this criterion, and high activity category otherwise. The classification method proceeds by computing the mean of each block and the number of pixels that have a specific distance from the mean. If the majority of the block pixels have less than the specific distance from the mean, the block is classified as being of the low activity type. There is a need for setting the specific distance and the majority number, and through testing some images these values are chosen to be one standard deviation and about 63 percent of all the block pixels for a block with 4x4 pixels. Despite the simplicity of this method, it proves to be very effective in distinguishing active regions of the image, such as edges, from the rest. Results of tests using some typical images are shown in Figure 8.3. Both the original and the detected high activity regions (white coloured areas) are shown in the figure.

The method has been explained using image pixels, but it can be used to detect the high activity codevectors as well. It is noted that for a universal codebook with a small size the method is not very effective, since for most of the codevectors, the components are close to the mean of the codevector. For a large codebook, the result obtained are in good agreement with those obtained in the pixel domain. Figures 8.4.a and 8.4.b depict the percentage differences between the results of the classification method in the pixel and the quantisation domains. The range of quantisation is from 5 bpv to 9 bpv. In Figure 8.5, the active blocks which are found in the quantisation domain for the test images *Lena* and *Peppers* are shown at 8 bpv.

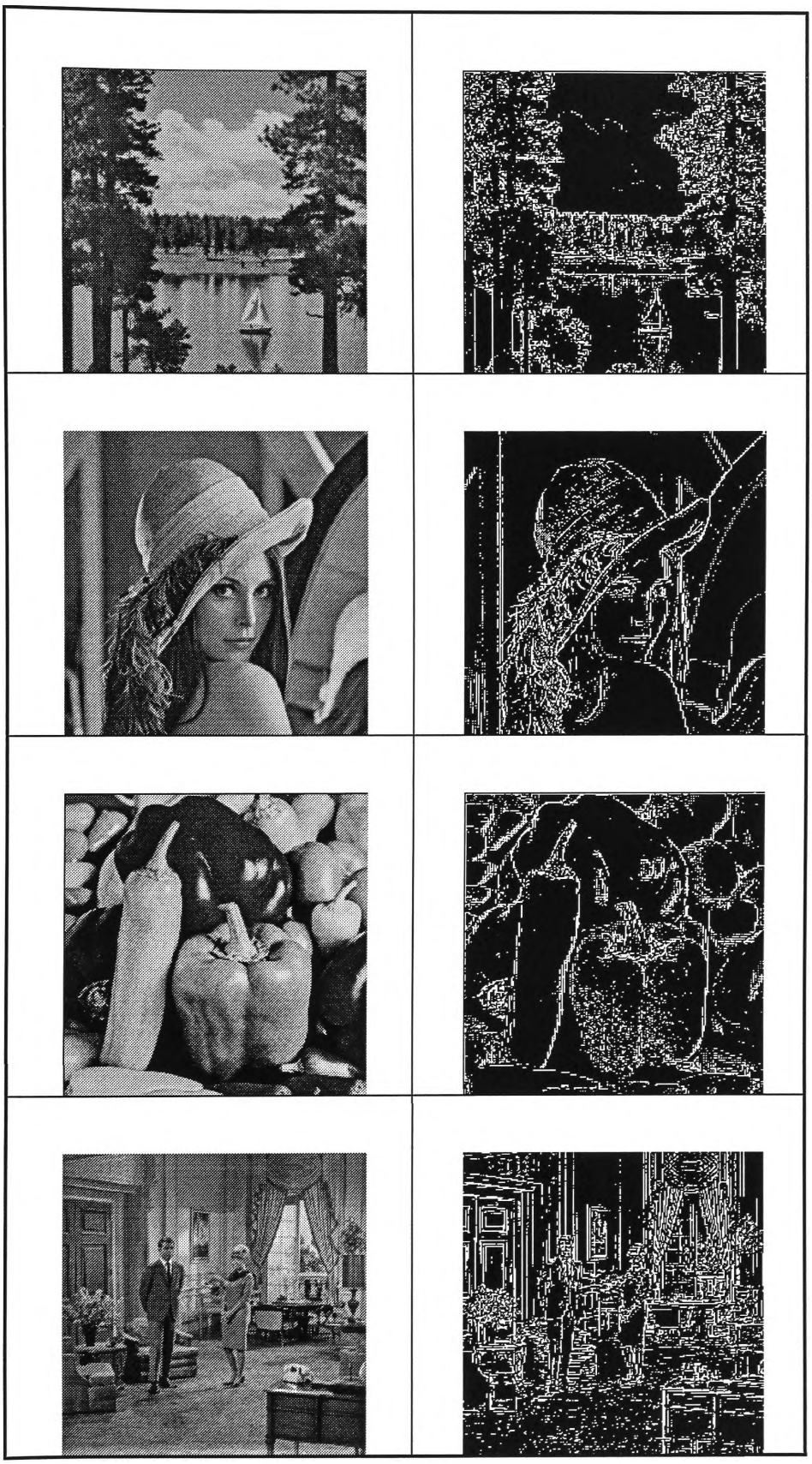

Figure 8.3: Detected active blocks

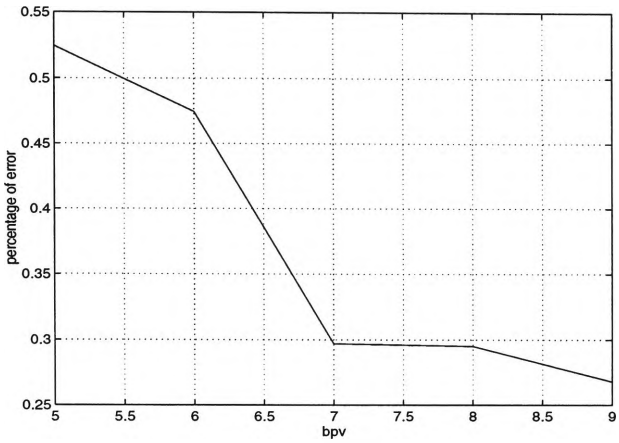

Figure 8.4.a: Error between finding the active blocks from the quantised and the original image (*Lena*)

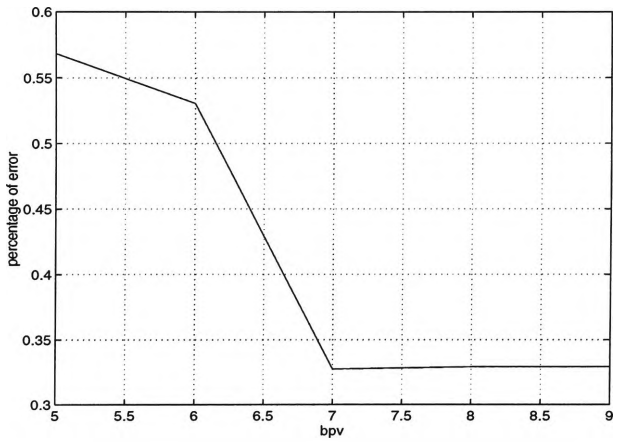

Figure 8.4.b: Error between finding the active blocks from the quantised and the original image (*Peppers*)

Figure 8.5: The found active blocks from the quantised image at 8 bpv

8.4 Simulation Results and Discussion

Two tests have been performed to evaluate the proposed image compression scheme. In the first simulation, a comparison with the result of IC-IAVQ presented in Chapter 7 is given in order to see how variable bit assignment can improve the performance. The codebook for encoding the residual codevectors is image-based and its size is 32, and 8 bpp is assigned for transmitting each of the components of the codevectors. The size of the universal codebook has been changed from 64 to 512. In the second simulation, the objective is to compare the results of using universal and image based codebook in encoding the residual vectors. The comparison have been made objectively through a PSNR versus bit rate graphs and subjectively by presenting the reconstructed images at about the same bit rate.

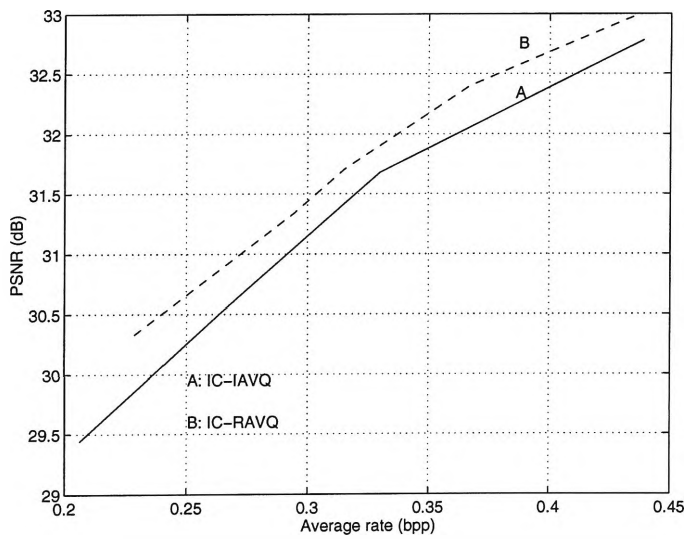

Figure 8.6: The results of IC-IAVQ and IC-RAVQ (*Lena*)

Figure 8.6 shows PSNR versus bit rate plots for IC-IAVQ (graph labelled "A") and IC-RAVQ (graph labelled "B") scheme using "*Lena*" as a test image. The improvement obtained by using IC-RAVQ is about 0.5 dB and

from a computational point of view, IC-RAVQ is also preferred to IC-IAVQ. Figures 8.7 and 8.8 show the images encoded by the two schemes, IC-IAVQ and IC-RAVQ, respectively. In terms of reproducing the edges and active areas such as around the eye or hair, IC-RAVQ has superior performance, because IC-RAVQ assigns more bits to those areas. IC-IAVQ is able to reproduce low activity regions better than IC-RAVQ. The reason is that, at the same bit rate, IC-IAVQ assigns more bits to low active areas than IC-RAVQ because of uniform bit allocation without any regard for the image block activity.

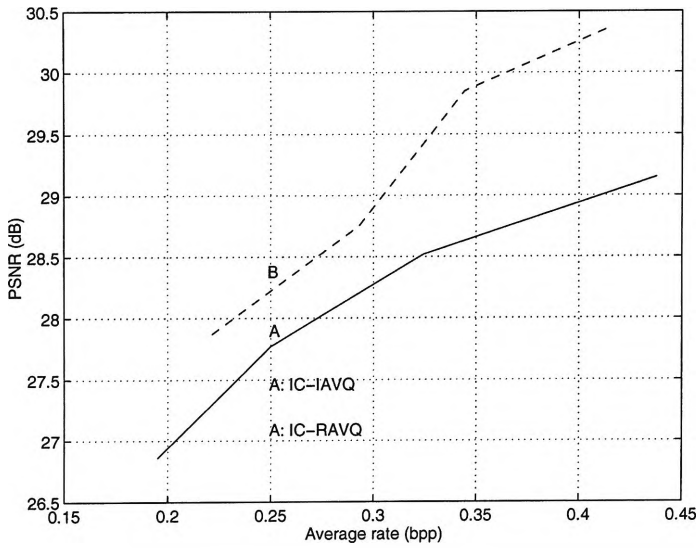

Figure 8.9: The results of IC-IAVQ and IC-RAVQ (*Peppers*)

Further tests using a more active image, *Peppers*, has been conducted to validate the veracity of the conclusion reached above. The PSNR versus bit rate plots for the two schemes, IC-IAVQ (graph labelled "A") and IC-RAVQ (graph labelled "B"), are shown in Figure 8.9. When the two graphs are compared, it is clear that IC-RAVQ has a performance advantage, about 1.25 dB on average, over IC-IAVQ. The reason is that high activity images contain more active blocks, consequently an adaptive bit allocation results in better performance than uniform bit allocation.

The subjective performance of the two methods are further compared by using the images shown in Figures 8.10 (IC-RAVQ) and 8.11 (IC-IAVQ). The edge reproduction capability of IC-RAVQ is obvious, especially when the edges of the big peppers in Figure 8.10 are considered. There is a possibility of the pixel values going out of range when the residual and the quantised versions are added. This may result in spotty artifacts in very dark or bright areas of the image. Some areas of the big black pepper exhibits this phenomenon. Note that this artifact can be easily corrected by taking the image through a rescaling post processing.

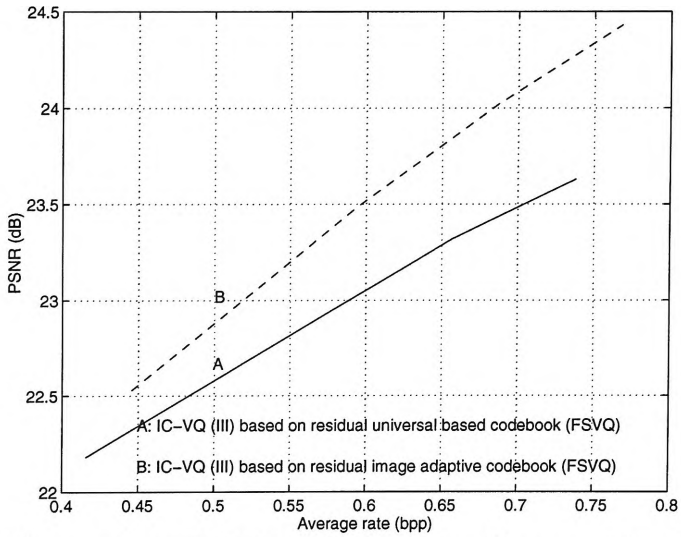

Figure 8.12. The results of IC-RAVQ in case of image adaptive codebook and universal based codebook for the residuals (*Baboon*)

The second set considered the effect of using a universal codebook as against an image-based codebook in encoding the residual vectors for the two test images. The results indicate a marginal difference both objectively and subjectively. Another test has been performed on a more active test image, *Baboon,* and the objective results (Figure 8.12) shows some differences. The reason adduced for this is that *Baboon* contains lots of variations which are unstructured and cannot be well reproduced by a universal based codebook, while in the case of *Lena* and *Peppers* the active areas are edges which can be found in most of natural images, consequently a universal-based codebook

for the residuals can reproduced those active blocks. Another reason for the marginal difference in performance between the two methods (for the residuals of the two test images, *Lena* and *Peppers*) is that at the same rate it is possible to choose a bigger codebook for the universal-based codebook, since there is no overhead to be incurred in transmitting the codebook. There is an exception in the case when image characteristics change gradually and the number of codevectors to be transmitted for the image based codebook is reduced: a better performance can be acheived by the image-based codebook in comparison with the universal-based codebook. In these circumstances both universal and image-based codebooks have about the same size and, the image-based codebook VQ coder results in less distortion.

8.5 Summary

This chapter addresses one of the problems of IAVQ; bit assignment procedure. In IAVQ all the image blocks are assigned bits uniformly without regard for their activities. This chapter introduces a residual adaptive VQ scheme based on a variable bit assignment depending the image blocks activity. This scheme first encodes the image by a universal VQ coder, then the residual of the active blocks derived from the difference between the quantised version and the original go into another quantisation process. The reason for the second process is to reproduce the active areas finely, at the expense of assigning more bits.

The objective and subjective quality of the new scheme has been compared to ordinary IC-IAVQ introduced in chapter 7. The results show that the new coder is able to reproduce the edges better than IC-IAVQ, and in terms of objective measurement presents a very good result for active images. In terms

Figure 8.7: Coded *Lena* by IC-RAVQ at 0.44 bpp and 33 dB PSNR

Figure 8.8: Coded *Lena* by IC-IAVQ at 0.44 bpp and 32.79 dB PSNR

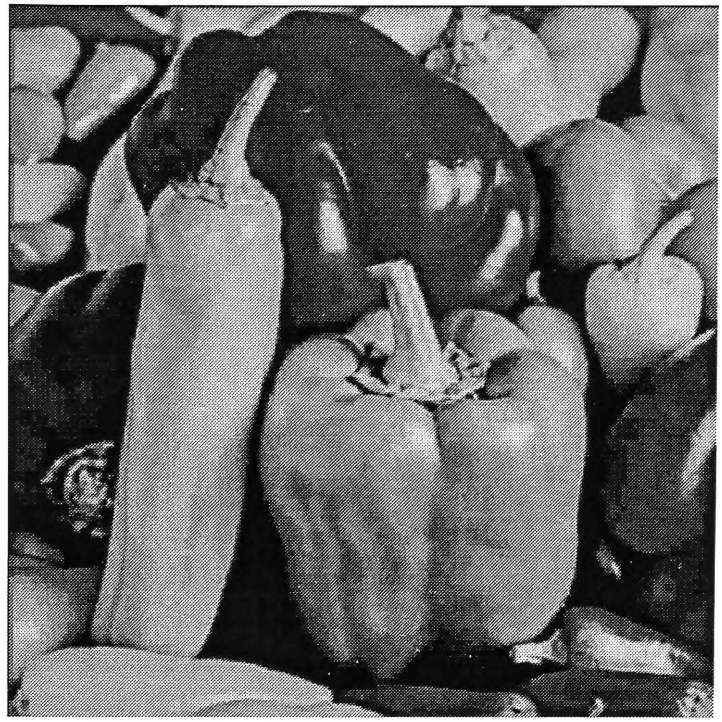

Figure 8.10: Coded *Peppers* by IC-RAVQ at 0.342 bpp and 29.21 PSNR

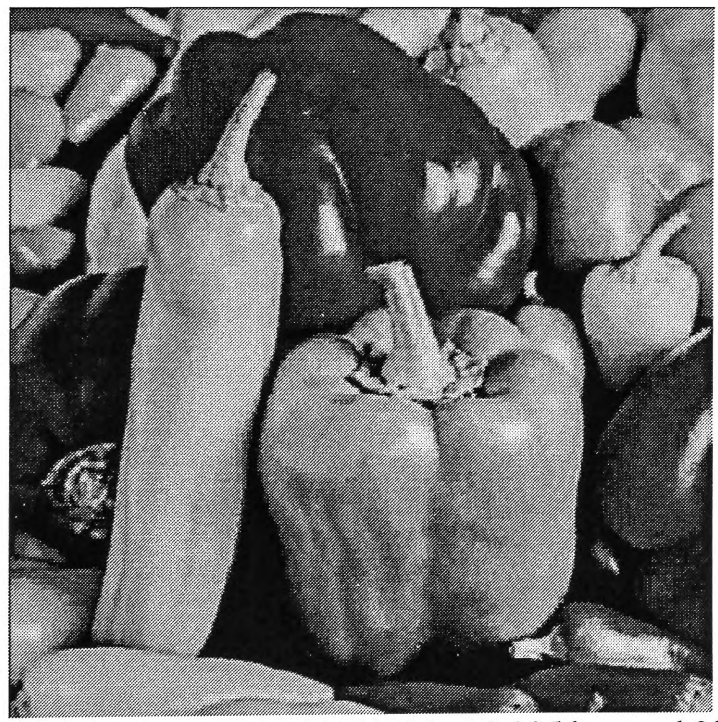

Figure 8.11: Coded *Peppers* by IC-IAVQ at 0.325 bpp and 28.52 PSNR

Chapter 9

Conclusion

9.1 Introduction

This thesis consists of four main parts: (i) literature review, (ii) the analogy between the index and pixel domains correlation, (iii) methods of index compression, and (iv) methods of improving the subjective quality of the coded image using index compression. Besides these four main points, which are presented in seven chapters, the first chapter presents a summary of what is going on in each chapter of the thesis and the main findings, and list of papers. Section 9.2 gives a brief summary of each of the four main points and the related contributions, and Section 9.3 offers suggestions for further research in the related areas of index compression.

9.2 Summary and Main Points

Chapter 2 begins with an outline where VQ fits into a bigger group of image compression techniques. VQ is considered as a block coding technique and its advantages and disadvantages in comparison with other block coding schemes such as transform and block truncation coding are discussed. The main advantage of VQ over TC and BTC is the simplicity of its decoder and low computational complexity at low bit rates over TC; its disadvantage is its poor coding performance at low bit rates.

A short literature review is next given on the methods of improving the performance of VQ at low bit rates, and the concept of index compression, which is the centre of attention in this thesis, is explained. Index compression is a method of exploiting the inter-block correlation, based on indices characteristics, to improve the bit rate (a lossless index compression method). Chapter 2 presents a discussion on a previous index compression method, address VQ; a brief description, its advantage and disadvantages are outlined.

Chapter 3 introduces the issue of analogy between the index and pixel domains correlation in highly correlated sources. This issue is the backbone of index compression schemes introduced in Chapters 4 to 6. Chapter 3 presents several tests on AR(I) and some natural images to show the high inter-index correlation and the similarity between the indices of adjacent image blocks or pixels. It also discusses the possibility of representing highly correlated sources, such as image data, by only the indices generated from quantising the data without any knowledge about the codebook.

Chapters 4 to 6 introduce three index compression methods. The basis of these three methods is that the indices of the vector quantised version of neighbouring vectors of highly correlated sources are mapped onto an original VQ indices subset. This fact is deduced from the results obtained in Chapter 3, Sections 5.2 and 6.2. Depending on whether a fixed or a variable size is used for the VQ indices subset, the three schemes can be categorised into two groups; Chapters 4 and 5 introduce methods for a fixed subset size, and Chapter 6 introduced the variable subset size case. The significant features of these methods are their simplicity, speed, low memory

requirement and comparable coding performance in comparison with address VQ. IC-VQ schemes are more sensitive to channel noise than the basic VQ and JPEG.

Chapter 4 presents an index compression method, IC-VQ (I), suitable for VQ coders when the codebook size is small. VQ coders with small codebook size results in high probability of identically indexed neighbouring blocks (or neighbouring blocks are mapped onto the VQ indices subset which have one member). A simple row and column processing on the indices of image blocks generates a map which shows the location of identically indexed neighbouring blocks and the rest. This map and the indices of those blocks which do not have identical index with any of their neighbours in the same row or column are sufficient to reconstruct all the image block indices. The necessary condition for performance improvement of IC-VQ (I) over its corresponding VQ coder is shown. This method results in up to 4 and 2 dB improvement over FSVQ and TSVQ, when the index compression is applied to the indices of FSVQ and TSVQ respectively.

Chapter 5 addresses the problem of IC-VQ (I) when the VQ codebook size become large. In this situation the probability of identically indexed neighbouring blocks is low, and consequently the rate required to represent those blocks which do not have identical indices with any of their preceding neighbours in the same row or column increases. Chapter 5 introduces a simple method, IC-VQ (II), to alleviate this problem by mapping the indices of neighbouring blocks on to a VQ indices subset which does not necessarily have one member as in IC-VQ (I). It is shown that the optimal size of the VQ indices subset depends on the corresponding VQ coder rate. IC-VQ (II) results

in 0.5 and 0.3 dB improvement over IC-VQ (I) based on FSVQ and TSVQ respectively. This improvement also includes allowance for large codebook size.

Chapter 6 presents an index compression scheme, IC-VQ (III) where instead of mapping the indices of neighbouring blocks onto a fixed size VQ indices subset, a variable size subset is considered. The significance of this scheme is that it allows for the non-stationarity of images; image areas possess varying levels of activity and thus requires varying sizes of codebook for representation. Low activity areas requires a smaller codebook size (less bits) than high activity areas, and a fixed size codebook results in allocating the same bit to low and high activity areas. The method introduced in Chapter 6 is primarily based on TSVQ, and is expanded to FSVQ by introducing the concept of virtual tree for a FSVQ. The variable bit assignment procedure of IC-VQ (III) results in its performance improvement over IC-VQ (II) of about 0.6 and 0.7 dB based on FSVQ and TSVQ respectively.

Index compression is a method which losslessly compresses the indices generated by a VQ scheme. Consequently this approach does not improve the subjective quality of the VQ coder at the same level of distortion. Chapters 7 and 8 deal with this problem. Chapter 7 presents the application of index compression in combination with image adaptive vector quantisation (IC-IAVQ). The results show significant improvement of up to 4 dB for images with low to moderate activity. It was shown that IC-IAVQ for high activity images not only outperforms basic IAVQ, but also outperforms the JPEG coding standard over a wide range of bit rates.

IAVQ assigns bits uniformly to image blocks disregarding their activity, consequently this deficiency is transferred to IC-IAVQ. Chapter 8 addresses this problem and introduces a method, IC-RAVQ, to solve it. Low activity blocks can be represented well by a universal-based codebook and high activity blocks require more bits for their representation. In IC-RAVQ, the image undergoes a VQ scheme and the resulting indices are transmitted. The residual of active blocks are further vector quantised and their indices are transmitted. A simple, but subjectively successful method is introduced to distinguish the active image blocks from the rest without any overhead information. The performance of IC-RAVQ shows significant improvement over IC-IAVQ for images with high activity. The implementation of IC-RAVQ is computationally more efficient than IC-IAVQ.

9.2 Suggestions for Further Research

There are at least three areas related to the work presented in this thesis in which further research can be conducted with the aim of improving the performance and extending the possible applications. Methods of further reducing the bit rate of index compression schemes will lead to an overall improvement of the compression schemes. The subjective quality of the decompressed image can become better by improving the bit allocation strategy. Lastly methods of applying these compression schemes to medical images may open the way to efficient lossless or near-lossless compression techniques for these types of images.

9.2.1 Bit rate reduction of index compression schemes

The index compression schemes introduced in this thesis are lossless schemes which require overhead information to represent the most similar

neighbouring blocks. There is the possibility of the existence of other methods for index compression and methods to reduce the overhead rate.

All the index compressed VQ schemes introduced are a combination of a VQ scheme and a lossless index compression scheme. The reason for this structure is to retain the performance of the original VQ coder. However, there are some lossy schemes that can provide significant results. A simple approach is to apply a VQ coder with small block size to compress the indices. Some of the results for small block size VQ coding have been shown in Appendix B. The problem associated with this simple approach is the possibility of misclassification because of loss which may result in a high distortion in active areas. An approach to compensate this problem is to use a method similar to the one proposed in Chapter 8.

One of the disadvantages of index compression schemes introduced in this thesis is the overhead required to represent which of the neighbours of a block in the north- or west-side has the most similarity with that block. There is a possibility to obtain these information adaptively. Because of high correlation in natural images, it is possible to predict the direction in which the most similarity exists, consequently eliminating the need to transmit any overhead information.

9.2.2 Index compression and bit allocation

Bit allocation can be considered from two points of view, how to allocate bits to minimise the overall distortion or how to allocate bits to minimise the subjectively noticeable distortion. Further research in these areas as related to index compression are considered.

Chapter 8 addressed the problem of uniform bit allocation of IAVQ by introducing a variable rate bit assignment to image blocks depending on their activity. This method assumes a universal or image-based codebook for the residual of active blocks, but this codebook was a fixed rate one, while active blocks have varying levels of activity, and bits should be allocated based on their activity. An approach to improve the method introduced in Chapter 8 is to generate a variable rate coder such as pruned tree- or greedy tree-structured VQ [Lookabaugh 1993, Riskin 1991].

The ultimate goal of any image compression scheme is to satisfy the human viewer. In situations where the subject under compression is a natural image, the subjective quality of the decompressed image is the most important factor in the quality of the coded image. In natural images, there are blocks with high activity which are not visually important, such as grasses, furs, and curly hair. These blocks require high amounts of bits for a good representation. A method to improve the subjective quality performance of the scheme proposed in Chapter 8 is to recognise these areas and avoid considering them in further bit allocation.

9.2.3 Application of Index Compression

This thesis presents the results of the index compression in combination with fixed rate VQ coders on natural images. Other areas of research are the application of index compression in combination with variable rate VQ coders, and some other types of images such as medical images.

The effectiveness of index compression methods on the other VQ schemes with memory such as predictive VQ, or FS-VQ, or variable rate coders such as pruned TSVQ and greedy tree structured VQ remains an open question.

Of course, with predictive VQ, or FS-VQ, it is obvious that as the correlation has been exploited, index compression cannot be as effective as it is for the memoryless VQ schemes.

Medical images contain very low activity areas, such as background, which can be compressed easily by index compression schemes. On the other hand, there are active areas in medical images that are the most important parts of the images, and cannot be easily compressed by index compression, unless a method similar to the scheme introduced in Chapter 8 is used.

References

S. Adlersberg S., V. Cuperman (1987), "Transform domain vector quantization for speech signals," ICASSP, Vol. 4, pp 45.44.1-45.4.4, Dallas, Texas, April 1987.

N. Ahmed, K. R. Rao (1975), "Orthogonal transforms for digital signal processing," New York: Springer, 1975.

K. Aizawa, H. Harashima, H. Miyakawa (1987), "Adaptive vector quantization of picture signals in discrete cosine transform domain," Elect. and Comm. in Japan, Part I, No. 70, 1987.

R. Aravind, A. Gersho, (1986) "Low-rate image coding with finite-state vector quantization," ICASSP, pp. 137-140, 1986.

R. Aravind, A. Gersho (1987), "Image compression based on vector quantization with finite state memory," Optical Engineering, Vol. 26, pp. 570-580, July 1987.

C. F. Barnes, R. L. Frost (1990), "Necessary conditions for the optimality of residual vector quantizer," Abstracts of 1990 IEEE International Symposium on Information Theory, pp. 34, San Diego, Calif., Jan. 1990.

J. L. Boxerman, H. J. Lee (1990), "Variable block-sized vector quantization of grayscale images with unconstrained tiling," ICASSP, pp. 2277-2280, 1990.

L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone (1984), "Classification and Regression Tree," Wadsworth, Belmont, Calif., 1984.

A. Buzo, A. H. Gray, Jr., R. M. Gray, J. D. Markel (1980), "Speech coding based upon vector quantization," IEEE Trans. Acoustic, Speech Signal Processing, vol. 28., pp. 562-574, Oct. 1980.

P. C. Chang, R. M. Gray (1986), "Gradient algorithms for designing predictive vector quantizers," IEEE Trans. Acoustic, Speech Signal Processing, ASSP-34, pp. 679-690, Aug. 1986.

O. T.-C. Chen, B. J. Sheu (1994), "An Adaptive Vector Quantizer Based on the Gold-Washing Method for Image Compression," IEEE Trans. on Circuits and Systems for Video Technology, Vol. 4. No. 2, pp. 143-157, April 1994.

W. Chen, C. H. Smith (1977), "Adaptive Coding of Monochrome and Color Images," IEEE Trans. on Comm., Vol. Com-25, No. 11, pp. 1285-1292, Nov. 1977.

W. Chen, W. K. Pratt (1984), "Scene Adaptive Coder," IEEE Trans. on Comm., Vol. Com-32, No. 3, pp. 225-232, March 1984.

W. Chen, E. Yang, Z. Zhang (1995), "A Variant of Address Vector Quantization for Image Compression Using Lossless Conditional Entropy Coding," ICASSP, pp 2483-6, 1995

P. A. Chou, T. Lookabaugh, R. M. Gray (1989), "Entropy-constrained vector quantization," IEEE Trans. on ASSP, pp. 31-42, Jan. 1989.

R. J. Clarke (1985), "Transform coding of images," Orlando, FL, Academic Press, 1985

C. Constantinescu, J. A. Storer (1994), "Improved Techniques for Single-Pass Adaptive Vector Quantization," Proceedings of the IEEE, Vol. 82, No. 6, pp. 933-939, June 1994.

P. C. Cosman (1993), "Perceptual aspects of vector quantization," Ph.D. dissertation, Stanford Univ., Stanford, CA, May 1993.

W. R. Daumer (1982), "Subjective Evaluation of Several Efficient Speech Coders," IEEE Trans. on Communications, pp. 655-662, April 1982.

E. J. Delp, Mitchell O. R. (1979), "Image Coding Using Block Truncation Coding," IEEE Trans. on Comm., Vol. 27, pp. 1335-1342, Sep. 1979.

I. Dinstein, K. Rose, A. Heiman (1990), "Variable Block-Size Transform Image Coder," IEEE Trans. on Comm., Vol. 38, No. 11, pp. 2073-2078, Nov. 1990.

O. Egger, Wei Li, M. Kunt (1995), "High Compression Image Coding Using an Adaptive Morphological Subband Decomposition," Proceedings of the IEEE, Vol. 83, No. 2, pp. 272-287, Feb. 1995.

W. Equitz (1989), "A New Vector Quantization Clustering Algorithm," IEEE Trans. Acoustic, Speech, Signal Processing, Vol. ASSP-37, No. 10, pp. 1568-1575, Oct. 1989.

Y. Feng, N. M. Nasrabadi (1988), "A dynamic address-vector quantization algorithm based on interblock and intercolor correlation for color image coding," ICASSP, pp. 1755-1758, May 1988.

J. Foster, R. M. Gray, M. O. Dunham (1985), "Finite-state vector quantization for waveform coding," IEEE Trans. Information Theory, vol. IT-31, pp. 348-359, May 1985.

P. Franti, O. Nevalainen, T. Kaukoranta (1994), "Compression of Digital Images by Block Truncation Coding: A Survey", The Computer Journal, Vol. 37, No. 4, pp. 308-331, 1994.

P. Franti, O. Nevalainen (1995), "Block Truncation Coding with Entropy Coding," IEEE Trans. on Communications., Vol. 43, No. 2/3/4, pp. 1677-1685, 1995.

R. L. Frost, C. F. Barnes, F. Xu (1991), "Design and performance of residual quantizers," IEEE Computer Society Press, Proceedings Data Compression Conference, pp. 129-138, Snowbird, Utah, April 1991.

A. Gersho, M. Yano (1985), "Adaptive Vector Quantization by Progressive Codevector Replacement," Proceedings IEEE International

Conference on Acoustic, Speech and Signal Processing (ICASSP), May 1985.

A. Gersho, R. M. Gray (1992), "Vector quantization and signal compression," Kluwer Academic Publisher, Massachusetts, 1992.

M. Golberg, P. R. Boutcher, S. Shlien (1986), "Image compression using adaptive vector quantisation," IEEE Tran. on Commun., Vol. Com 34., pp. 180_187, Feb. 1986.

R. M. Gray (1984), "Vector quantization," IEEE ASSP Mag., pp. 4-29, Apr. 1984.

D. Healy, O. R. Mitchell (1981), "Digital Bandwidth Compression Using Block Truncation Coding," IEEE Trans. on Comm., Vol. 29, pp. 1809-1817, 1981.

J. J. Y. Huang, P. M. Schultheiss, P. M. (1963), "Block quantisation of correlate Gaussian random variables," IEEE Trans. Comm. Systems Com-11, pp. 289-296, 1963.

Shih-Chi Huang, Yip-Fang Huang, (1994), "A Constrained Vector Quantization for Real-Time Codebook Retransmission," IEEE Trans. on Circuits and Systems for Video Technology, Vol. 4. No. 1. , pp. 1-7, Feb. 1994.

D. A. Huffman (1952), "A method for the constructure of minimum redundancy codes," Proceedings of the IRE, No. 40, pp. 1098-1101, 1952.

A. K. Jain (1989), "Fundamental of Digital Image Processing," Englewood Cliffs, NJ: Prentice-Hall, 1989.

A. K. Jain (1981), "Image data compression: A review," Proceedings of the IEEE, Vol. 69, No. 3, March 1981.

N. S. Jayant, J. Johnston, R. Safranek (1993), "Signal Compression Based on Models Of Human Perception," Proceedings of the IEEE, Vol. 81, No. 10, pp. 1385-1422, Oct. 1993.

N. S. Jayant, P. Noll (1984), "Digital Coding of Wave forms," Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

C. B. Jones (1981), "An efficient coding system for long source sequences," IEEE Trans. Information Theory, IT-27, pp. 280-291, 1981.

B. H. Juang, A. H. Gray (1982), "Multi stage vector quantization for speech coding," In international Conference on Acoustics Speech and Signal Processing, Volume 1, pp. 597-600, Paris, April 1982.

M. Kamel, C. Sun, L. Guan (1991), "Image Compression by Variable Block Truncation Coding with Optimal Threshold," IEEE Trans. on Signal Processing, Vol. 39, pp. 208-212, 1991.

D. S. Kim, S. U. Lee (1991), "Image Vector Quantizer Based on a Classification in the DCT Domain," IEEE Trans. on Communications, Vol. 39, No. 4, pp. 549-556, April 1991.

J. W. Kim, S. U. Lee (1992), "A Transform Domain Classified Vector Quantizer for Image Coding," IEEE Trans. on Circuits and Systems for Video Technology, Vol. 2, No. 1, pp. 3-14, March 1992.

T. Kim (1988), "New finite state vector quantizers for images," ICASSP, pp. 1180-1183, 1988.

R. A. King, N. M. Nasrabadi (1983), "Image coding using vector quantization in the transform domain," Pattern Recognition Letters, No. 1, pp. 323-329, 1983.

M. Kunt, A. Ikonomopoulos, M. Kocher (1985), "Second-generation image-coding techniques," Proceeding of the IEEE, Vol. 73, No. 4, April 1985.

Y. Linde, A. Bozo, R. M. Gray (1980), "An algorithm for vector quantizer design," IEEE Trans. Commun., vol. COM-28, pp. 84-95, Jan. 1980.

T. Lookabaugh, E. A. Riskin, P. A. Chou, R. M. Gray (1993), "Variable Rate Vector Quantization for Speech, Image, and Video Compression," IEEE Trans. on Comm., Vol. 41, No. 1, Jan. 1993.

W. W. Lu, M. Gough, P. Davies (1991), "Scientific Data Compression for Space: a modified Block Truncation Coding Algorithm," Proc. SPIE, 1470, pp. 197-205, 1991.

J. Makhoul, S. Roucos and H. Gish (1985), "Vector quantization in speech coding," Proc. IEEE, Vol. 73, pp. 1551-1588, Nov. 1985.

H. B. Mitchell, E. J. Delp (1980), "Multilevel Graphics Representation Using Block Truncation Coding," Proceeding of the IEEE, Vol. 68, pp. 868-873, 1980.

M. Miyahara (1985), "Block Distortion in Orthogonal Transform Coding-Analysis, Minimization, and Distortion Measure," IEEE Trans. on Comm. Vol. COM-33, No. 1, pp. 90-96, Jan. 1985.

P. Nasiopoulos, R. Ward, D. Morse (1991), "Adaptive Compression Coding," IEEE Trans. on Communications, Vol. 39, pp. 1245-1254, 1991.

N. M. Nasrabadi, Y. Feng (1988), "Vector Quantization of Images Based Upon the Kohonen Self-Organizing Feature Maps," in Proc. 2nd ICNN Conf., Vol. I, pp. 101-105, 1988.

N. M. Nasrabadi, R. A. King (1988), "Image coding using vector quantization: A review," IEEE Trans. on Communications, COM-35, pp. 957-971, Aug. 1988.

N. M. Nasrabadi, Y. Feng (1990a), "A Multilayer Address Vector Quantization Technique," IEEE Trans. on Circuits and Systems, Vol. 37, No. 7, pp. 912-921, July 1990.

N. M. Nasrabadi., Y. Feng (1990b), "Image compression using address-vector quantisation," IEEE Trans. Comm., Vol. 38, pp. 2166-2173, Dec. 1990.

A. N. Netravali (1977), "On Quantizers for DPCM Coding of Picture Signals," IEEE Trans. on Information Theory IT-23, pp. 360-370, May 1977.

A. N. Netravali, B. G. Haskell (1989), "Digital Pictures: Representation and Compression," New York: Plenum, 1989.

A. N. Netravali, J. O. Limb (1980), "Picture Coding: A Review," Proceedings of the IEEE, Vol 68, No. 3, pp. 366-406, March 1980.

S. Panchanathan, M. Goldberg (1991), "Mini-max Algorithm for Image Adaptive Vector quantisation," IEE Proceedings-1, Vol. 138, No. 1, pp. 53-60, Feb 1991.

H. Park, V. K. Prasanna (1993), "Modular VLSI Architectures for Real-Time Full-Search-Based Vector Quantization," IEEE Trans. on Circuits and Systems for Video Technology, Vol. 3, No. 4, pp. 309-317, Aug. 1993.

R. Pasco (1976), "Source coding algorithms for fast data compression," Ph. D. Dissertation, Stanford University, 1976.

W. A. Pearlman, P. Jakatdar, M. M. Leung (1992), "Adaptive Transform Tree Coding of Images," IEEE Journal on selected areas in Comm., Vol. 10, No. 5, pp. 902-912, June 1992.

W. B. Pennebaker, J. L. Mitchell (1993), "JPEG Still Image Data Compression Standard," Van Nostrand Reinhold, New York, 1993.

B. Ramamurthi, A. Gersho (1986), "Classified vector quantization of images," IEEE Trans. on Comm., COM-34, pp. 1105-1115, Nov. 1986.

K. R. Rao, Y. Yip (1990), "Discrete Cosine Transform, Algorithms, Advantages, Applications," Academic Press, San Diego, Calif., 1990.

E. A. Riskin, T. Lookabaugh, P. A. Chou, and R. M. Gray (1990), "Variable rate vector quantization for medical image compression," IEEE Trans. on Medical Imaging, Vol. 9, pp. 290-298, Sep. 1990.

E. A. Riskin, R. M. Gray (1991), "A greedy tree growing algorithm for the design of variable rate vector quantizers," IEEE Trans. Signal Process., Vol. 39, No 11, pp. 2500-2507, Nov. 1991.

J. Rissanen, G. G. Langdon (1981), "Universal modeling and coding," IEEE Trans. Information Theory, IT-27, 1981.

J. Roy, M. N. Nasrabadi (1991), "Hierarchical Block Truncation Coding," Optical Engineering, Vol. 30, pp. 551-556, 1991.

H. Samet (1984), "The quad-tree and related hierarchical data structures," ACM Computing Surveys, Vol. 16, No. 2, pp. 188-260, June 1984.

V. Udpikar, J. Raina (1987), "BTC Image Coding Using Vector Quantization," IEEE Trans. on Comm., Vol. 35, pp. 352-356, 1987.

J. Vaisey, A. Gersho (1988), "Variable block-size image coding," ICASSP, pp. 1051-1054, 1988.

J. Vaisey, A. Gersho (1992), "Image Compression with Variable Block Size Segmentation," IEEE Trans. Signal Process., Vol. 40, No. 8, pp. 2040-2060, Aug. 1992.

G. K. Wallace (1992), "The JPEG Still Picture Compression Standard," pp. 18-34, IEEE Tran Consumer Electronics, vol. 38, No. 1, Feb. 1992.

L. Wang, M. Golberg, S. Shlien (1992), "Unified Variable-Length Transform Coding and Image Adaptive Vector Quantisation," IEEE Journal on selected areas in communication, vol. 10. No. 5, pp. 892-901, June 1992.

Li Weiping, Ya-Qin Zhang (1995), "Vector-Based Signal Processing and Quantization for Image and Video Compression," Proceeding of the IEEE, Vol. 83, No. 2, pp. 317-334, Feb. 1995.

Y. Wu, D. C. Coll (1991), "BTC-VQ-DCT Hybrid Coding of Digital Images," IEEE Trans. on Comm., Vol 39, pp. 1283-1287, 1991.

Y. Wu, D. C. Coll (1993), "Multilevel Block Truncation Coding of Color Images," IEEE Journal on Selected Areas in Comm., Vol. 10, pp. 952-959, 1993.

K. Zeger, A. Bist (1992), "Universal Adaptive Vector Quantization with Application to Image Compression," Proc. ICASP, San Francisco, Calif., March 1992.

K. Zeger, A. Bist, T. Linder (1994), "Universal Source Coding with Codebook Transmission," IEEE Trans. on Comm., Vol. 42, No. 2/3/4, 1994.

Appendices

Appendix A: The Probability of Identically Indexed Neighbouring Blocks

1. Results for 2x2 Block Size

1.1 Results for TSVQ

Table 1.1.1: The probability of having an identical index with the neighbouring block located in the north-west-side

	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
Airplane	0.4477	0.5155	0.5551	0.6398	0.7287	0.8541	0.9318
Announcer	0.4052	0.4647	0.5405	0.6564	0.7639	0.8633	0.9565
Baboon	0.0851	0.1228	0.1699	0.2600	0.4141	0.6017	0.8062
Boat	0.2676	0.3270	0.3927	0.5226	0.6377	0.7750	0.9094
Cablecar	0.3800	0.4221	0.5010	0.5882	0.6856	0.8109	0.9345
Cornfield	0.3811	0.4148	0.4549	0.5079	0.6124	0.7384	0.8461
Couple	0.2321	0.2884	0.3426	0.4219	0.5970	0.7580	0.8745
Crowd	0.2250	0.2798	0.3596	0.4683	0.5817	0.7526	0.9062
Flower	0.4538	0.5051	0.5587	0.6305	0.7554	0.8444	0.9375
Fruits	0.4425	0.5123	0.5547	0.5973	0.7471	0.8499	0.9010
Girl	0.5373	0.6065	0.6693	0.7331	0.8343	0.9144	0.9671
Hustler	0.4702	0.5254	0.5842	0.6425	0.7639	0.8623	0.9359
Lena	0.3528	0.4017	0.4640	0.5589	0.6865	0.8311	0.9238
Light	0.4819	0.5536	0.6002	0.6171	0.7359	0.8802	0.9409
Man	0.2580	0.3115	0.3868	0.4857	0.6236	0.7799	0.9012
Masuda1	0.5557	0.6248	0.6568	0.7303	0.8440	0.9094	0.9560
Masuda2	0.6103	0.6788	0.7031	0.7713	0.8718	0.9292	0.9630
Model	0.5595	0.6147	0.6419	0.6667	0.8087	0.9087	0.9507
Pens	0.4117	0.4539	0.5120	0.5685	0.7092	0.8368	0.9552
Peppers	0.3235	0.4077	0.4914	0.6289	0.7486	0.8617	0.9410
Soccer	0.2614	0.3068	0.3462	0.3959	0.5597	0.7114	0.8357
Stdimg	0.4413	0.5566	0.6149	0.7280	0.8476	0.9396	0.9801
Tanaka	0.5337	0.6230	0.6750	0.7356	0.8435	0.9216	0.9640
Woman1	0.3398	0.3917	0.4337	0.5399	0.7115	0.8136	0.9309
Woman2	0.4196	0.4914	0.5687	0.6690	0.7928	0.9071	0.9682
Yatch	0.3574	0.3973	0.4354	0.5020	0.6391	0.7670	0.8873

Table 1.1.2: The probability of having an identical index with the neighbouring block located in the north-side

	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
Airplane	0.4907	0.5614	0.6009	0.6830	0.7762	0.8901	0.9514
Announcer	0.4763	0.5379	0.6115	0.7139	0.8088	0.8907	0.9652
Baboon	0.1079	0.1518	0.2061	0.3000	0.4570	0.6367	0.8201
Boat	0.3071	0.3746	0.4483	0.5725	0.6899	0.8179	0.9296
Cablecar	0.4206	0.4698	0.5514	0.6319	0.7317	0.8458	0.9488
Cornfield	0.3952	0.4310	0.4758	0.5258	0.6336	0.7652	0.8697
Couple	0.3532	0.4181	0.4807	0.5549	0.7115	0.8383	0.9185
Crowd	0.3036	0.3623	0.4463	0.5470	0.6631	0.8100	0.9307
Flower	0.5189	0.5733	0.6275	0.6935	0.8070	0.8847	0.9547
Fruits	0.5079	0.5750	0.6202	0.6646	0.8000	0.8854	0.9268
Girl	0.6088	0.6749	0.7310	0.7846	0.8680	0.9321	0.9733
Hustler	0.5531	0.6095	0.6646	0.7189	0.8184	0.8991	0.9553
Lena	0.4649	0.5230	0.5933	0.6846	0.7956	0.8998	0.9563
Light	0.5088	0.5804	0.6254	0.6424	0.7566	0.8989	0.9517
Man	0.3421	0.4031	0.4802	0.5732	0.7099	0.8404	0.9302
Masuda1	0.6217	0.6854	0.7174	0.7773	0.8742	0.9281	0.9674
Masuda2	0.6766	0.7438	0.7694	0.8277	0.9094	0.9537	0.9778
Model	0.6890	0.7273	0.7538	0.7757	0.8687	0.9422	0.9698
Pens	0.3778	0.4178	0.4807	0.5408	0.6979	0.8340	0.9536
Peppers	0.3840	0.4771	0.5636	0.6957	0.8043	0.9040	0.9628
Soccer	0.2949	0.3448	0.3905	0.4408	0.6075	0.7556	0.8620
Stdimg	0.5264	0.6425	0.7082	0.8094	0.9037	0.9637	0.9892
Tanaka	0.5997	0.6878	0.7410	0.7950	0.8842	0.9441	0.9721
Woman1	0.4010	0.4590	0.5086	0.6152	0.7760	0.8650	0.9532
Woman2	0.5063	0.5730	0.6481	0.7376	0.8423	0.9335	0.9783
Yatch	0.4359	0.4810	0.5251	0.5861	0.7124	0.8266	0.9191

Table 1.1.3: The probability of having an identical index with the neighbouring block located in the west-side

	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
Airplane	0.5171	0.5899	0.6263	0.7045	0.7850	0.8952	0.9534
Announcer	0.5201	0.5891	0.6584	0.7639	0.8534	0.9236	0.9804
Baboon	0.1081	0.1533	0.2045	0.3080	0.4768	0.6681	0.8446
Boat	0.3177	0.3802	0.4436	0.5716	0.6923	0.8180	0.9309
Cablecar	0.4477	0.4948	0.5736	0.6588	0.7577	0.8658	0.9587
Cornfield	0.4405	0.4791	0.5209	0.5775	0.6806	0.7979	0.8915
Couple	0.3293	0.3939	0.4514	0.5352	0.6904	0.8184	0.9068
Crowd	0.2980	0.3570	0.4357	0.5418	0.6600	0.8090	0.9332
Flower	0.5208	0.5763	0.6276	0.6965	0.8079	0.8863	0.9581
Fruits	0.5177	0.5880	0.6304	0.6726	0.8035	0.8834	0.9248
Girl	0.5876	0.6590	0.7203	0.7832	0.8732	0.9391	0.9780
Hustler	0.5373	0.5953	0.6497	0.7115	0.8216	0.9046	0.9572
Lena	0.3766	0.4302	0.4936	0.5894	0.7168	0.8517	0.9360
Light	0.6087	0.6715	0.7104	0.7296	0.8317	0.9281	0.9653
Man	0.3078	0.3678	0.4420	0.5401	0.6778	0.8213	0.9229
Masuda1	0.6306	0.6979	0.7291	0.7977	0.8943	0.9433	0.9759
Masuda2	0.6422	0.7086	0.7314	0.7951	0.8878	0.9385	0.9707
Model	0.5954	0.6500	0.6764	0.7011	0.8335	0.9236	0.9599
Pens	0.4155	0.4636	0.5262	0.5917	0.7427	0.8624	0.9567
Peppers	0.3647	0.4568	0.5403	0.6720	0.7866	0.8893	0.9564
Soccer	0.3816	0.4391	0.4817	0.5352	0.6857	0.8105	0.9030
Stdimg	0.5148	0.6126	0.6711	0.7784	0.8854	0.9566	0.9851
Tanaka	0.5995	0.6852	0.7363	0.7919	0.8847	0.9444	0.9770
Woman1	0.3649	0.4191	0.4632	0.5716	0.7392	0.8351	0.9410
Woman2	0.4871	0.5583	0.6297	0.7245	0.8376	0.9286	0.9759
Yatch	0.5054	0.5551	0.5945	0.6572	0.7780	0.8631	0.9372

Table 1.1.4: The probability of having identical an index with any of the neighbouring blocks located in the north- or west-side

	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
Airplane	0.6650	0.7261	0.7588	0.8219	0.8918	0.9571	0.9833
Announcer	0.6809	0.7499	0.8052	0.8772	0.9357	0.9720	0.9944
Baboon	0.1859	0.2524	0.3317	0.4689	0.6609	0.8244	0.9328
Boat	0.4491	0.5265	0.6085	0.7203	0.8339	0.9233	0.9783
Cablecar	0.5553	0.6127	0.6885	0.7662	0.8577	0.9344	0.9821
Cornfield	0.5035	0.5505	0.6043	0.6718	0.7841	0.8967	0.9581
Couple	0.5261	0.6010	0.6715	0.7563	0.8715	0.9422	0.9749
Crowd	0.4435	0.5124	0.5961	0.6940	0.8102	0.9124	0.9751
Flower	0.6819	0.7349	0.7743	0.8267	0.9037	0.9532	0.9843
Fruits	0.6650	0.7295	0.7740	0.8164	0.9072	0.9531	0.9713
Girl	0.7713	0.8264	0.8646	0.8991	0.9482	0.9790	0.9927
Hustler	0.7098	0.7646	0.8070	0.8564	0.9245	0.9669	0.9883
Lena	0.5795	0.6402	0.7066	0.7851	0.8720	0.9440	0.9785
Light	0.7486	0.7992	0.8293	0.8451	0.9187	0.9724	0.9879
Man	0.4828	0.5513	0.6296	0.7240	0.8441	0.9294	0.9745
Masuda1	0.7880	0.8378	0.8661	0.9054	0.9580	0.9815	0.9922
Masuda2	0.7969	0.8484	0.8719	0.9074	0.9590	0.9841	0.9933
Model	0.8086	0.8403	0.8635	0.8858	0.9420	0.9795	0.9908
Pens	0.5614	0.6168	0.6828	0.7491	0.8782	0.9502	0.9901
Peppers	0.5348	0.6344	0.7134	0.8154	0.8950	0.9537	0.9866
Soccer	0.4840	0.5549	0.6104	0.6763	0.8127	0.9134	0.9639
Stdimg	0.6915	0.7729	0.8331	0.9016	0.9559	0.9871	0.9967
Tanaka	0.7474	0.8175	0.8626	0.9016	0.9523	0.9802	0.9911
Woman1	0.5445	0.6061	0.6594	0.7588	0.8763	0.9351	0.9802
Woman2	0.6877	0.7500	0.8039	0.8672	0.9360	0.9761	0.9929
Yatch	0.6268	0.6821	0.7273	0.7839	0.8817	0.9445	0.9809

Table 1.1.5: The probability of having an identical index with any of the neighbouring blocks located in the north-, west- or north-west-side

	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
Airplane	0.7023	0.7536	0.7861	0.8454	0.9082	0.9628	0.9846
Announcer	0.7164	0.7795	0.8283	0.8915	0.9428	0.9752	0.9948
Baboon	0.2255	0.3010	0.3945	0.5462	0.7321	0.8714	0.9513
Boat	0.4962	0.5761	0.6579	0.7631	0.8644	0.9379	0.9802
Cablecar	0.5864	0.6451	0.7187	0.7966	0.8793	0.9448	0.9847
Cornfield	0.5288	0.5792	0.6369	0.7125	0.8206	0.9177	0.9647
Couple	0.5618	0.6349	0.7047	0.7876	0.8893	0.9511	0.9782
Crowd	0.4800	0.5520	0.6343	0.7301	0.8370	0.9260	0.9773
Flower	0.7209	0.7668	0.8014	0.8489	0.9173	0.9596	0.9851
Fruits	0.7008	0.7604	0.8032	0.8450	0.9215	0.9600	0.9752
Girl	0.8088	0.8540	0.8862	0.9149	0.9570	0.9820	0.9933
Hustler	0.7440	0.7941	0.8319	0.8766	0.9354	0.9713	0.9895
Lena	0.6216	0.6795	0.7425	0.8140	0.8913	0.9522	0.9813
Light	0.7852	0.8295	0.8561	0.8715	0.9321	0.9747	0.9883
Man	0.5291	0.5969	0.6756	0.7671	0.8701	0.9405	0.9776
Masuda1	0.8159	0.8596	0.8846	0.9192	0.9628	0.9831	0.9926
Masuda2	0.8235	0.8673	0.8894	0.9208	0.9653	0.9863	0.9938
Model	0.8312	0.8595	0.8813	0.9040	0.9510	0.9825	0.9918
Pens	0.6472	0.7012	0.7578	0.8168	0.9138	0.9621	0.9914
Peppers	0.5914	0.6815	0.7537	0.8419	0.9099	0.9590	0.9875
Soccer	0.5181	0.5896	0.6491	0.7189	0.8394	0.9260	0.9667
Stdimg	0.7290	0.8013	0.8548	0.9114	0.9590	0.9877	0.9968
Tanaka	0.7811	0.8429	0.8818	0.9152	0.9582	0.9817	0.9916
Woman1	0.6029	0.6616	0.7148	0.8044	0.9000	0.9472	0.9831
Woman2	0.7335	0.7909	0.8364	0.8897	0.9453	0.9787	0.9933
Yatch	0.6489	0.7036	0.7483	0.8047	0.8950	0.9511	0.9827

Table 1.1.6: The probability of having an identical index with any of the neighbouring blocks located in the north-, west-, north-west- or north-east-side

	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
Airplane	0.7409	0.7892	0.8225	0.8770	0.9344	0.9772	0.9920
Announcer	0.7537	0.8139	0.8587	0.9135	0.9578	0.9835	0.9968
Baboon	0.2668	0.3513	0.4594	0.6189	0.7986	0.9132	0.9683
Boat	0.5406	0.6234	0.7070	0.8046	0.8975	0.9583	0.9871
Cablecar	0.6270	0.6878	0.7612	0.8355	0.9112	0.9643	0.9906
Cornfield	0.5487	0.6029	0.6655	0.7455	0.8527	0.9406	0.9765
Couple	0.5995	0.6727	0.7434	0.8255	0.9159	0.9671	0.9865
Crowd	0.5372	0.6127	0.6962	0.7862	0.8866	0.9570	0.9885
Flower	0.7727	0.8163	0.8488	0.8902	0.9477	0.9785	0.9928
Fruits	0.7539	0.8089	0.8512	0.8902	0.9495	0.9749	0.9848
Girl	0.8533	0.8917	0.9182	0.9417	0.9728	0.9895	0.9961
Hustler	0.7836	0.8302	0.8647	0.9051	0.9542	0.9815	0.9943
Lena	0.6854	0.7445	0.8069	0.8716	0.9369	0.9791	0.9933
Light	0.8216	0.8618	0.8858	0.9007	0.9503	0.9845	0.9925
Man	0.5860	0.6561	0.7366	0.8253	0.9132	0.9648	0.9886
Masuda1	0.8518	0.8901	0.9127	0.9410	0.9745	0.9891	0.9958
Masuda2	0.8567	0.8966	0.9177	0.9440	0.9777	0.9928	0.9970
Model	0.8664	0.8917	0.9124	0.9348	0.9697	0.9906	0.9963
Pens	0.6792	0.7342	0.7896	0.8467	0.9318	0.9717	0.9936
Peppers	0.6511	0.7379	0.8066	0.8835	0.9411	0.9781	0.9959
Soccer	0.5500	0.6243	0.6883	0.7592	0.8717	0.9471	0.9768
Stdimg	0.7847	0.8524	0.9078	0.9519	0.9804	0.9939	0.9987
Tanaka	0.8320	0.8854	0.9203	0.9471	0.9765	0.9905	0.9950
Woman1	0.6652	0.7258	0.7811	0.8620	0.9395	0.9727	0.9925
Woman2	0.7888	0.8416	0.8828	0.9274	0.9692	0.9898	0.9973
Yatch	0.6858	0.7397	0.7840	0.8374	0.9183	0.9659	0.9895

1.2 Results for FSVQ

Table 1.2.1: The probability of having an identical index with the neighbouring block located in the north-west-side

	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
Airplane	0.4128	0.4669	0.5504	0.6333	0.7204	0.8472	0.9246
Announcer	0.3714	0.4416	0.5376	0.6367	0.7617	0.8508	0.9487
Baboon	0.0677	0.1032	0.1732	0.2544	0.4052	0.5933	0.8004
Boat	0.2290	0.2936	0.3999	0.5111	0.6373	0.7720	0.9020
Cablecar	0.3590	0.4120	0.5005	0.5695	0.6757	0.7978	0.9178
Cornfield	0.3728	0.3809	0.4660	0.5112	0.6172	0.7338	0.8438
Couple	0.1582	0.2058	0.3325	0.4074	0.5850	0.7472	0.8676
Crowd	0.1884	0.2334	0.3374	0.4354	0.5761	0.7471	0.8981
Flower	0.3981	0.4549	0.5543	0.6185	0.7468	0.8443	0.9304
Fruits	0.3814	0.4307	0.5486	0.6084	0.7421	0.8417	0.8985
Girl	0.4672	0.5190	0.6547	0.7143	0.8339	0.9060	0.9544
Hustler	0.4224	0.4905	0.6018	0.6655	0.7834	0.8785	0.9351
Lena	0.2615	0.3209	0.4512	0.5317	0.6761	0.8232	0.9166
Light	0.3926	0.4610	0.5942	0.6227	0.7440	0.8746	0.9339
Man	0.1853	0.2470	0.3531	0.4363	0.6125	0.7696	0.8947
Masuda1	0.5240	0.5499	0.6788	0.7406	0.8335	0.8946	0.9425
Masuda2	0.5546	0.5890	0.7152	0.7790	0.8601	0.9158	0.9486
Model	0.4666	0.5301	0.6468	0.6739	0.7908	0.8845	0.9395
Pens	0.3909	0.4410	0.5362	0.5982	0.7234	0.8370	0.9414
Peppers	0.2924	0.3746	0.5007	0.6239	0.7418	0.8549	0.9333
Soccer	0.2465	0.2685	0.3775	0.4242	0.5683	0.7087	0.8366
Stdimg	0.4263	0.4951	0.6142	0.7254	0.8372	0.9325	0.9723
Tanaka	0.4799	0.5223	0.6263	0.7071	0.8129	0.8962	0.9554
Woman1	0.2511	0.3167	0.4384	0.5312	0.6977	0.8043	0.9243
Woman2	0.3369	0.4182	0.5596	0.6618	0.7857	0.8998	0.9607
Yatch	0.3508	0.3812	0.4854	0.5062	0.6422	0.7553	0.8756

Table 1.2.2: The probability of having an identical index with the neighbouring block located in the north-side

	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
Airplane	0.4553	0.5125	0.5970	0.6795	0.7672	0.8827	0.9439
Announcer	0.4666	0.5370	0.6305	0.7172	0.8215	0.8953	0.9614
Baboon	0.0895	0.1337	0.2082	0.2941	0.4501	0.6291	0.8133
Boat	0.2699	0.3427	0.4546	0.5653	0.6897	0.8131	0.9220
Cablecar	0.4238	0.4787	0.5673	0.6332	0.7379	0.8463	0.9428
Cornfield	0.4117	0.4255	0.5110	0.5521	0.6582	0.7786	0.8775
Couple	0.2867	0.3437	0.4739	0.5502	0.7005	0.8293	0.9119
Crowd	0.2652	0.3213	0.4295	0.5269	0.6589	0.8049	0.9230
Flower	0.4775	0.5367	0.6370	0.7004	0.8070	0.8815	0.9491
Fruits	0.4648	0.5178	0.6333	0.6950	0.8119	0.8867	0.9277
Girl	0.5541	0.6083	0.7314	0.7806	0.8673	0.9263	0.9658
Hustler	0.5389	0.6020	0.7030	0.7538	0.8358	0.9081	0.9551
Lena	0.3773	0.4463	0.5860	0.6628	0.7860	0.8911	0.9488
Light	0.4265	0.4896	0.6290	0.6534	0.7655	0.8930	0.9449
Man	0.2643	0.3364	0.4561	0.5385	0.6990	0.8313	0.9230
Masuda1	0.6007	0.6284	0.7362	0.7899	0.8708	0.9217	0.9611
Masuda2	0.6431	0.6725	0.7810	0.8342	0.9029	0.9461	0.9697
Model	0.6120	0.6692	0.7614	0.7843	0.8666	0.9375	0.9616
Pens	0.3699	0.4231	0.5192	0.5825	0.7224	0.8434	0.9443
Peppers	0.3559	0.4452	0.5770	0.6906	0.7976	0.8970	0.9551
Soccer	0.2949	0.3252	0.4358	0.4829	0.6273	0.7643	0.8730
Stdimg	0.5120	0.5847	0.7049	0.8036	0.8941	0.9564	0.9815
Tanaka	0.5695	0.6137	0.7184	0.7861	0.8758	0.9354	0.9678
Woman1	0.3061	0.3794	0.5133	0.6059	0.7645	0.8562	0.9457
Woman2	0.4205	0.5038	0.6401	0.7290	0.8345	0.9258	0.9708
Yatch	0.4487	0.4835	0.5874	0.6088	0.7307	0.8331	0.9164

Table 1.2.3: The probability of having an identical index with the neighbouring block located in the west-side

	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
Airplane	0.4821	0.5420	0.6302	0.6987	0.7798	0.8874	0.9460
Announcer	0.4651	0.5395	0.6404	0.7286	0.8342	0.9010	0.9681
Baboon	0.0939	0.1424	0.2214	0.2995	0.4692	0.6610	0.8380
Boat	0.2734	0.3451	0.4547	0.5585	0.6918	0.8137	0.9236
Cablecar	0.4216	0.4822	0.5713	0.6378	0.7463	0.8508	0.9410
Cornfield	0.4276	0.4497	0.5360	0.5782	0.6837	0.7901	0.8851
Couple	0.2466	0.3124	0.4475	0.5180	0.6799	0.8093	0.8997
Crowd	0.2514	0.3108	0.4217	0.5138	0.6546	0.8033	0.9250
Flower	0.4545	0.5180	0.6193	0.6785	0.7946	0.8830	0.9495
Fruits	0.4427	0.5023	0.6198	0.6731	0.7959	0.8732	0.9198
Girl	0.5115	0.5686	0.7057	0.7616	0.8712	0.9311	0.9652
Hustler	0.4748	0.5463	0.6627	0.7244	0.8347	0.9139	0.9521
Lena	0.2809	0.3437	0.4810	0.5609	0.7067	0.8433	0.9285
Light	0.5007	0.5852	0.7001	0.7277	0.8342	0.9208	0.9579
Man	0.2237	0.2989	0.4090	0.4936	0.6661	0.8109	0.9162
Masuda1	0.5881	0.6226	0.7482	0.8043	0.8813	0.9271	0.9609
Masuda2	0.5831	0.6224	0.7407	0.8010	0.8770	0.9254	0.9568
Model	0.4943	0.5653	0.6791	0.7028	0.8130	0.8987	0.9493
Pens	0.3745	0.4425	0.5437	0.6132	0.7508	0.8580	0.9436
Peppers	0.3342	0.4220	0.5473	0.6664	0.7789	0.8825	0.9487
Soccer	0.3395	0.3870	0.4973	0.5476	0.6819	0.7997	0.8938
Stdimg	0.4953	0.5645	0.6661	0.7643	0.8757	0.9491	0.9773
Tanaka	0.5348	0.5874	0.6935	0.7653	0.8558	0.9198	0.9681
Woman1	0.2697	0.3419	0.4671	0.5608	0.7267	0.8266	0.9342
Woman2	0.3952	0.4843	0.6249	0.7181	0.8306	0.9211	0.9685
Yatch	0.4780	0.5317	0.6309	0.6502	0.7681	0.8418	0.9218

Table 1.2.4: The probability of having an identical index with any of the neighbouring blocks located in the north- or west-side

	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
Airplane	0.6255	0.6850	0.7623	0.8166	0.8838	0.9487	0.9754
Announcer	0.6414	0.7161	0.8069	0.8673	0.9298	0.9655	0.9862
Baboon	0.1628	0.2398	0.3510	0.4586	0.6531	0.8168	0.9249
Boat	0.4031	0.4982	0.6156	0.7148	0.8297	0.9165	0.9704
Cablecar	0.5346	0.5992	0.6925	0.7591	0.8552	0.9292	0.9747
Cornfield	0.5092	0.5502	0.6358	0.6864	0.7964	0.8967	0.9552
Couple	0.4438	0.5335	0.6743	0.7473	0.8625	0.9342	0.9675
Crowd	0.3920	0.4701	0.5922	0.6834	0.8054	0.9063	0.9673
Flower	0.6242	0.6872	0.7786	0.8257	0.8989	0.9473	0.9771
Fruits	0.5966	0.6650	0.7775	0.8243	0.9085	0.9489	0.9666
Girl	0.7077	0.7666	0.8608	0.8928	0.9428	0.9716	0.9848
Hustler	0.6801	0.7443	0.8330	0.8716	0.9286	0.9645	0.9825
Lena	0.4918	0.5731	0.6997	0.7680	0.8633	0.9354	0.9709
Light	0.6491	0.7143	0.8195	0.8449	0.9175	0.9648	0.9803
Man	0.3866	0.4813	0.6144	0.6985	0.8338	0.9208	0.9668
Masuda1	0.7540	0.7910	0.8719	0.9069	0.9513	0.9737	0.9847
Masuda2	0.7574	0.7957	0.8721	0.9080	0.9529	0.9766	0.9855
Model	0.7359	0.7912	0.8623	0.8825	0.9354	0.9728	0.9828
Pens	0.5108	0.5934	0.7029	0.7688	0.8821	0.9474	0.9813
Peppers	0.4966	0.5981	0.7226	0.8099	0.8881	0.9467	0.9788
Soccer	0.4368	0.5026	0.6315	0.6863	0.8146	0.9097	0.9607
Stdimg	0.6631	0.7354	0.8212	0.8879	0.9475	0.9796	0.9888
Tanaka	0.7035	0.7578	0.8466	0.8937	0.9450	0.9717	0.9846
Woman1	0.4392	0.5303	0.6684	0.7476	0.8660	0.9270	0.9726
Woman2	0.5983	0.6830	0.8028	0.8614	0.9282	0.9683	0.9851
Yatch	0.6102	0.6695	0.7669	0.7907	0.8830	0.9393	0.9746

Table 1.2.5: The probability of having an identical index with any of the neighbouring blocks located in the north-, west- or north-west-side

	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
Airplane	0.6605	0.7189	0.7904	0.8397	0.8996	0.9544	0.9767
Announcer	0.6767	0.7484	0.8298	0.8822	0.9363	0.9684	0.9866
Baboon	0.1988	0.2885	0.4149	0.5355	0.7222	0.8633	0.9439
Boat	0.4510	0.5514	0.6664	0.7597	0.8596	0.9312	0.9724
Cablecar	0.5617	0.6282	0.7222	0.7895	0.8752	0.9387	0.9771
Cornfield	0.5296	0.5770	0.6646	0.7220	0.8271	0.9151	0.9601
Couple	0.4807	0.5748	0.7088	0.7781	0.8813	0.9432	0.9705
Crowd	0.4304	0.5123	0.6338	0.7210	0.8321	0.9193	0.9695
Flower	0.6662	0.7228	0.8048	0.8474	0.9127	0.9535	0.9779
Fruits	0.6326	0.7005	0.8062	0.8480	0.9200	0.9549	0.9698
Girl	0.7521	0.8058	0.8816	0.9087	0.9511	0.9746	0.9853
Hustler	0.7174	0.7777	0.8542	0.8874	0.9373	0.9680	0.9835
Lena	0.5409	0.6240	0.7366	0.7994	0.8829	0.9442	0.9738
Light	0.6934	0.7507	0.8466	0.8697	0.9294	0.9670	0.9808
Man	0.4346	0.5337	0.6637	0.7454	0.8612	0.9322	0.9700
Masuda1	0.7845	0.8189	0.8879	0.9182	0.9558	0.9750	0.9851
Masuda2	0.7903	0.8260	0.8902	0.9205	0.9586	0.9786	0.9859
Model	0.7685	0.8183	0.8796	0.8998	0.9445	0.9757	0.9837
Pens	0.5997	0.6793	0.7767	0.8296	0.9136	0.9578	0.9826
Peppers	0.5510	0.6515	0.7620	0.8358	0.9027	0.9519	0.9798
Soccer	0.4668	0.5373	0.6674	0.7238	0.8391	0.9217	0.9625
Stdimg	0.7003	0.7665	0.8435	0.9021	0.9507	0.9802	0.9889
Tanaka	0.7414	0.7909	0.8685	0.9080	0.9504	0.9734	0.9849
Woman1	0.5042	0.5979	0.7236	0.7947	0.8904	0.9392	0.9755
Woman2	0.6558	0.7332	0.8349	0.8836	0.9373	0.9709	0.9855
Yatch	0.6295	0.6896	0.7848	0.8093	0.8944	0.9450	0.9757

Table 1.2.6: The probability of having an identical index with any of the neighbouring blocks located in the north-, west-, north-west- or north-east-side

	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
Airplane	0.6986	0.7583	0.8260	0.8713	0.9251	0.9690	0.9841
Announcer	0.7149	0.7854	0.8600	0.9045	0.9502	0.9761	0.9886
Baboon	0.2360	0.3395	0.4769	0.6071	0.7878	0.9047	0.9607
Boat	0.4967	0.6002	0.7132	0.8018	0.8922	0.9511	0.9794
Cablecar	0.6004	0.6699	0.7617	0.8278	0.9051	0.9571	0.9827
Cornfield	0.5475	0.6000	0.6900	0.7520	0.8563	0.9361	0.9709
Couple	0.5214	0.6202	0.7487	0.8162	0.9090	0.9593	0.9790
Crowd	0.4861	0.5759	0.6964	0.7825	0.8816	0.9498	0.9806
Flower	0.7196	0.7738	0.8482	0.8874	0.9426	0.9721	0.9852
Fruits	0.6883	0.7569	0.8533	0.8898	0.9461	0.9686	0.9781
Girl	0.8010	0.8483	0.9119	0.9339	0.9660	0.9821	0.9881
Hustler	0.7568	0.8166	0.8819	0.9115	0.9522	0.9758	0.9874
Lena	0.6088	0.6967	0.8002	0.8596	0.9290	0.9715	0.9857
Light	0.7367	0.7888	0.8793	0.8990	0.9470	0.9767	0.9849
Man	0.4936	0.6004	0.7294	0.8078	0.9051	0.9569	0.9809
Masuda1	0.8199	0.8554	0.9121	0.9368	0.9665	0.9802	0.9876
Masuda2	0.8274	0.8632	0.9159	0.9408	0.9705	0.9844	0.9887
Model	0.8109	0.8581	0.9084	0.9280	0.9621	0.9831	0.9883
Pens	0.6297	0.7109	0.8028	0.8545	0.9280	0.9656	0.9847
Peppers	0.6098	0.7124	0.8134	0.8771	0.9332	0.9706	0.9881
Soccer	0.4970	0.5736	0.7016	0.7604	0.8682	0.9407	0.9710
Stdimg	0.7559	0.8179	0.8923	0.9410	0.9727	0.9863	0.9908
Tanaka	0.7939	0.8424	0.9088	0.9382	0.9675	0.9818	0.9880
Woman1	0.5703	0.6683	0.7898	0.8529	0.9308	0.9649	0.9847
Woman2	0.7186	0.7918	0.8812	0.9210	0.9612	0.9817	0.9895
Yatch	0.6662	0.7279	0.8165	0.8403	0.9149	0.9585	0.9815

2 Results for 4x4 block size

2.1 Results for TSVQ

Table 2.1.1: The probability of having an identical index with the neighbouring block located in the north-west-side

	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
Airplane	0.3816	0.4444	0.4815	0.5771	0.6646	0.8087	0.9080
Announcer	0.3722	0.4304	0.4647	0.5817	0.6930	0.8042	0.9325
Baboon	0.0927	0.1333	0.1761	0.2721	0.4300	0.6357	0.8176
Boat	0.2753	0.3137	0.3491	0.4553	0.5695	0.7141	0.8766
Cablecar	0.3771	0.4062	0.4613	0.5452	0.6530	0.7941	0.9307
Cornfield	0.3709	0.4157	0.4340	0.4728	0.5808	0.7097	0.8291
Couple	0.2069	0.2715	0.2978	0.3685	0.5290	0.7065	0.8426
Crowd	0.1767	0.2101	0.2506	0.3485	0.4749	0.6678	0.8696
Flower	0.4091	0.4679	0.4849	0.5426	0.6733	0.7834	0.9020
Fruits	0.3996	0.4621	0.4800	0.5779	0.7089	0.8335	0.8906
Girl	0.4931	0.5615	0.5843	0.6681	0.7721	0.8703	0.9514
Hustler	0.4117	0.4736	0.4914	0.5746	0.6708	0.7893	0.9016
Lena	0.3076	0.3682	0.3973	0.4930	0.6062	0.7658	0.8883
Light	0.4834	0.5525	0.5570	0.5943	0.6919	0.8332	0.9113
Man	0.2169	0.2725	0.3052	0.3878	0.5503	0.7182	0.8613
Masuda1	0.5047	0.5584	0.6038	0.6726	0.7676	0.8589	0.9340
Masuda2	0.5395	0.5920	0.6486	0.7212	0.8133	0.8908	0.9422
Model	0.4358	0.5073	0.5170	0.6623	0.7234	0.8690	0.9272
Pens	0.3691	0.4195	0.4435	0.5528	0.6557	0.7916	0.9416
Peppers	0.3406	0.3883	0.4466	0.5642	0.6694	0.7955	0.9092
Soccer	0.2148	0.2711	0.2891	0.3397	0.5033	0.6492	0.8006
Stdimg	0.4414	0.4678	0.5388	0.6544	0.7787	0.9039	0.9634
Tanaka	0.5170	0.5581	0.5693	0.6759	0.7831	0.8875	0.9477
Woman1	0.3040	0.3793	0.3964	0.4899	0.6374	0.7555	0.9017
Woman2	0.3682	0.4262	0.4798	0.5844	0.7021	0.8502	0.9437
Yatch	0.3296	0.3719	0.3896	0.4789	0.5997	0.7317	0.8694

Table 2.A.2: The probability of having an identical index with the neighbouring block located in the north-side

	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
Airplane	0.4130	0.4844	0.5264	0.6269	0.7144	0.8486	0.9344
Announcer	0.4439	0.5067	0.5404	0.6468	0.7464	0.8382	0.9457
Baboon	0.1300	0.1789	0.2244	0.3296	0.4894	0.6811	0.8417
Boat	0.3446	0.3901	0.4288	0.5324	0.6335	0.7739	0.9055
Cablecar	0.4090	0.4416	0.4969	0.5849	0.6942	0.8216	0.9417
Cornfield	0.3747	0.4215	0.4426	0.4853	0.6001	0.7302	0.8449
Couple	0.3606	0.4293	0.4615	0.5314	0.6683	0.8079	0.8983
Crowd	0.2476	0.2879	0.3374	0.4451	0.5673	0.7419	0.9038
Flower	0.4623	0.5287	0.5492	0.6088	0.7285	0.8313	0.9278
Fruits	0.4460	0.5118	0.5326	0.6339	0.7604	0.8619	0.9089
Girl	0.5808	0.6412	0.6625	0.7374	0.8204	0.9007	0.9609
Hustler	0.5220	0.5797	0.5977	0.6671	0.7515	0.8456	0.9321
Lena	0.4600	0.5213	0.5591	0.6470	0.7502	0.8650	0.9401
Light	0.5391	0.6022	0.6070	0.6498	0.7292	0.8556	0.9265
Man	0.3107	0.3736	0.4175	0.5051	0.6508	0.7965	0.9046
Masuda1	0.5730	0.6284	0.6590	0.7228	0.8081	0.8855	0.9477
Masuda2	0.6177	0.6677	0.7173	0.7822	0.8647	0.9260	0.9639
Model	0.6156	0.6753	0.6863	0.7645	0.8181	0.9146	0.9553
Pens	0.2922	0.3450	0.3682	0.4860	0.5997	0.7630	0.9294
Peppers	0.4321	0.4861	0.5456	0.6521	0.7497	0.8607	0.9447
Soccer	0.2319	0.2916	0.3156	0.3721	0.5399	0.6879	0.8241
Stdimg	0.5676	0.5988	0.6548	0.7552	0.8534	0.9422	0.9813
Tanaka	0.5934	0.6378	0.6538	0.7450	0.8366	0.9183	0.9595
Woman1	0.3775	0.4608	0.4820	0.5807	0.7214	0.8293	0.9331
Woman2	0.4749	0.5362	0.5851	0.6820	0.7789	0.8957	0.9639
Yatch	0.3974	0.4421	0.4646	0.5605	0.6757	0.8052	0.9102

Table 2.A.3: The probability of having an identical index with the neighbouring block located in the west-side

	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
Airplane	0.4892	0.5490	0.5862	0.6678	0.7424	0.8620	0.9352
Announcer	0.5468	0.5948	0.6288	0.7244	0.8169	0.8980	0.9709
Baboon	0.1127	0.1590	0.2046	0.3072	0.4823	0.6809	0.8425
Boat	0.3353	0.3814	0.4155	0.5166	0.6338	0.7677	0.9070
Cablecar	0.4417	0.4776	0.5232	0.6065	0.7230	0.8473	0.9541
Cornfield	0.4469	0.4838	0.5079	0.5499	0.6593	0.7722	0.8760
Couple	0.3154	0.3854	0.4148	0.4869	0.6356	0.7825	0.8870
Crowd	0.2388	0.2799	0.3238	0.4227	0.5488	0.7281	0.9004
Flower	0.4768	0.5391	0.5539	0.6126	0.7371	0.8380	0.9343
Fruits	0.4748	0.5384	0.5560	0.6477	0.7706	0.8692	0.9169
Girl	0.5616	0.6370	0.6607	0.7349	0.8240	0.9085	0.9679
Hustler	0.5400	0.5961	0.6162	0.6881	0.7761	0.8711	0.9398
Lena	0.3391	0.4016	0.4308	0.5229	0.6323	0.7879	0.9037
Light	0.5864	0.6561	0.6659	0.7020	0.7902	0.8983	0.9487
Man	0.2631	0.3268	0.3612	0.4427	0.6100	0.7705	0.8925
Masuda1	0.5864	0.6408	0.6861	0.7579	0.8432	0.9131	0.9636
Masuda2	0.5679	0.6192	0.6721	0.7415	0.8336	0.9048	0.9554
Model	0.4762	0.5513	0.5631	0.6995	0.7629	0.8923	0.9419
Pens	0.3362	0.3984	0.4225	0.5327	0.6544	0.8052	0.9382
Peppers	0.3892	0.4395	0.4951	0.6074	0.7128	0.8330	0.9295
Soccer	0.3382	0.4019	0.4243	0.4802	0.6320	0.7606	0.8800
Stdimg	0.4859	0.5140	0.5721	0.6894	0.8257	0.9300	0.9725
Tanaka	0.5845	0.6294	0.6423	0.7371	0.8379	0.9216	0.9693
Woman1	0.3323	0.4101	0.4264	0.5191	0.6715	0.7897	0.9187
Woman2	0.4478	0.5110	0.5620	0.6550	0.7598	0.8874	0.9580
Yatch	0.4644	0.5148	0.5395	0.6201	0.7277	0.8218	0.9172

Table 2.A.4: The probability of having an identical index with any of the neighbouring blocks located in the north- or west-side

	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
Airplane	0.6110	0.6746	0.7128	0.7870	0.8585	0.9370	0.9763
Announcer	0.6930	0.7445	0.7795	0.8520	0.9146	0.9590	0.9916
Baboon	0.2064	0.2790	0.3505	0.4915	0.6907	0.8511	0.9346
Boat	0.4789	0.5374	0.5817	0.6800	0.7876	0.8961	0.9696
Cablecar	0.5336	0.5796	0.6326	0.7250	0.8308	0.9234	0.9794
Cornfield	0.4960	0.5372	0.5751	0.6404	0.7610	0.8767	0.9483
Couple	0.5315	0.6093	0.6518	0.7285	0.8443	0.9317	0.9716
Crowd	0.3638	0.4220	0.4832	0.5981	0.7263	0.8667	0.9633
Flower	0.6202	0.6854	0.7091	0.7641	0.8551	0.9240	0.9736
Fruits	0.6078	0.6813	0.7049	0.7895	0.8881	0.9430	0.9626
Girl	0.7491	0.8060	0.8248	0.8721	0.9245	0.9678	0.9896
Hustler	0.7150	0.7669	0.7870	0.8392	0.9043	0.9539	0.9843
Lena	0.5584	0.6204	0.6611	0.7423	0.8325	0.9213	0.9696
Light	0.7087	0.7739	0.7860	0.8280	0.8922	0.9572	0.9817
Man	0.4306	0.5124	0.5653	0.6596	0.7982	0.9035	0.9623
Masuda1	0.7482	0.8019	0.8238	0.8742	0.9318	0.9683	0.9873
Masuda2	0.7458	0.7906	0.8210	0.8723	0.9330	0.9730	0.9895
Model	0.7388	0.7988	0.8130	0.8707	0.9192	0.9695	0.9876
Pens	0.4553	0.5356	0.5701	0.6930	0.8079	0.9215	0.9876
Peppers	0.5716	0.6289	0.6846	0.7720	0.8527	0.9310	0.9797
Soccer	0.4170	0.4958	0.5344	0.6102	0.7612	0.8783	0.9535
Stdimg	0.6893	0.7240	0.7642	0.8446	0.9216	0.9763	0.9940
Tanaka	0.7322	0.7791	0.7985	0.8670	0.9280	0.9722	0.9878
Woman1	0.5193	0.6064	0.6340	0.7198	0.8409	0.9185	0.9727
Woman2	0.6412	0.7101	0.7478	0.8212	0.8965	0.9599	0.9881
Yatch	0.5799	0.6377	0.6727	0.7573	0.8523	0.9290	0.9751

Table 2.2.5: The probability of having an identical index with any of the neighbouring blocks located in the north-, west- or north-west-side

	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
Airplane	0.6422	0.7019	0.7390	0.8091	0.8786	0.9467	0.9791
Announcer	0.7225	0.7719	0.8069	0.8712	0.9255	0.9633	0.9920
Baboon	0.2472	0.3300	0.4194	0.5737	0.7621	0.8930	0.9533
Boat	0.5117	0.5709	0.6192	0.7192	0.8223	0.9150	0.9732
Cablecar	0.5640	0.6135	0.6725	0.7652	0.8597	0.9379	0.9835
Cornfield	0.5158	0.5617	0.6066	0.6819	0.8010	0.9071	0.9612
Couple	0.5566	0.6349	0.6809	0.7589	0.8663	0.9412	0.9748
Crowd	0.3939	0.4580	0.5251	0.6401	0.7663	0.8890	0.9676
Flower	0.6603	0.7184	0.7429	0.7934	0.8748	0.9344	0.9755
Fruits	0.6484	0.7222	0.7482	0.8226	0.9079	0.9536	0.9694
Girl	0.7792	0.8293	0.8464	0.8880	0.9355	0.9708	0.9902
Hustler	0.7396	0.7889	0.8082	0.8566	0.9136	0.9592	0.9855
Lena	0.5913	0.6504	0.6933	0.7705	0.8570	0.9337	0.9739
Light	0.7341	0.7976	0.8106	0.8513	0.9088	0.9626	0.9826
Man	0.4714	0.5559	0.6139	0.7070	0.8289	0.9180	0.9674
Masuda1	0.7771	0.8249	0.8454	0.8898	0.9410	0.9716	0.9883
Masuda2	0.7792	0.8197	0.8483	0.8923	0.9422	0.9763	0.9896
Model	0.7701	0.8224	0.8368	0.8881	0.9326	0.9758	0.9898
Pens	0.5560	0.6384	0.6785	0.7839	0.8750	0.9476	0.9902
Peppers	0.6124	0.6665	0.7196	0.7983	0.8702	0.9382	0.9809
Soccer	0.4523	0.5321	0.5772	0.6595	0.8000	0.9009	0.9599
Stdimg	0.7174	0.7517	0.7940	0.8662	0.9318	0.9784	0.9941
Tanaka	0.7635	0.8064	0.8263	0.8876	0.9387	0.9756	0.9889
Woman1	0.5760	0.6574	0.6891	0.7661	0.8703	0.9323	0.9765
Woman2	0.6806	0.7436	0.7778	0.8451	0.9116	0.9645	0.9888
Yatch	0.6047	0.6650	0.7020	0.7819	0.8691	0.9379	0.9780

Table 2.2.6: The probability of having an identical index with any of the neighbouring blocks located in the north-, west-, north-west-, or north-east-side

	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
Airplane	0.6782	0.7370	0.7767	0.8432	0.9105	0.9675	0.9896
Announcer	0.7591	0.8068	0.8409	0.8976	0.9438	0.9747	0.9939
Baboon	0.2943	0.3880	0.4897	0.6544	0.8290	0.9303	0.9700
Boat	0.5494	0.6127	0.6667	0.7665	0.8617	0.9402	0.9823
Cablecar	0.6001	0.6545	0.7189	0.8099	0.8964	0.9578	0.9884
Cornfield	0.5307	0.5808	0.6325	0.7174	0.8348	0.9320	0.9722
Couple	0.5898	0.6685	0.7193	0.7984	0.8979	0.9592	0.9838
Crowd	0.4418	0.5135	0.5901	0.7085	0.8289	0.9311	0.9827
Flower	0.7133	0.7703	0.7986	0.8494	0.9160	0.9632	0.9881
Fruits	0.7013	0.7752	0.8032	0.8700	0.9418	0.9704	0.9802
Girl	0.8282	0.8710	0.8873	0.9212	0.9568	0.9817	0.9938
Hustler	0.7713	0.8197	0.8405	0.8865	0.9369	0.9734	0.9922
Lena	0.6575	0.7178	0.7634	0.8335	0.9104	0.9689	0.9917
Light	0.7717	0.8316	0.8467	0.8874	0.9362	0.9764	0.9880
Man	0.5291	0.6175	0.6808	0.7754	0.8825	0.9518	0.9834
Masuda1	0.8119	0.8579	0.8761	0.9155	0.9578	0.9815	0.9933
Masuda2	0.8179	0.8552	0.8801	0.9194	0.9617	0.9868	0.9954
Model	0.8041	0.8565	0.8719	0.9159	0.9557	0.9854	0.9946
Pens	0.5819	0.6653	0.7076	0.8084	0.8937	0.9585	0.9927
Peppers	0.6654	0.7218	0.7728	0.8439	0.9094	0.9653	0.9944
Soccer	0.4791	0.5620	0.6133	0.7045	0.8361	0.9249	0.9704
Stdimg	0.7744	0.8110	0.8502	0.9117	0.9626	0.9879	0.9972
Tanaka	0.8160	0.8563	0.8772	0.9246	0.9635	0.9887	0.9941
Woman1	0.6390	0.7217	0.7593	0.8323	0.9219	0.9674	0.9899
Woman2	0.7437	0.8042	0.8346	0.8930	0.9462	0.9834	0.9961
Yatch	0.6388	0.6963	0.7357	0.8130	0.8926	0.9540	0.9857

2.2 Results for FSVQ

Table 2.2.1: The probability of having an identical index with the neighbouring block located in the north-west-side

	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
Airplane	0.3932	0.4196	0.5285	0.5718	0.6570	0.7947	0.8937
Announcer	0.3168	0.3771	0.4998	0.5420	0.6796	0.7791	0.9181
Baboon	0.0743	0.1101	0.2046	0.2482	0.4184	0.6196	0.8042
Boat	0.2156	0.2678	0.3917	0.4511	0.5755	0.7081	0.8622
Cablecar	0.3245	0.3600	0.4614	0.5124	0.6244	0.7634	0.9048
Cornfield	0.3450	0.3621	0.4433	0.4623	0.5719	0.6899	0.8114
Couple	0.1477	0.1873	0.3203	0.3386	0.5125	0.6915	0.8282
Crowd	0.1179	0.1562	0.2732	0.3250	0.4666	0.6572	0.8552
Flower	0.3275	0.3674	0.4848	0.5069	0.6587	0.7698	0.8869
Fruits	0.3306	0.3730	0.5211	0.5264	0.6825	0.8070	0.8753
Girl	0.4247	0.4615	0.6063	0.6134	0.7587	0.8540	0.9286
Hustler	0.3651	0.4065	0.5476	0.5525	0.6858	0.7875	0.8921
Lena	0.2542	0.2969	0.4301	0.4535	0.5911	0.7506	0.8739
Light	0.4237	0.4152	0.6083	0.5787	0.6727	0.8232	0.8970
Man	0.1633	0.2235	0.3358	0.3576	0.5323	0.6973	0.8479
Masuda1	0.4290	0.4740	0.6094	0.6323	0.7382	0.8231	0.9056
Masuda2	0.4792	0.5233	0.6592	0.6953	0.7857	0.8632	0.9155
Model	0.3849	0.4275	0.5767	0.5274	0.6946	0.8300	0.9058
Pens	0.3425	0.3862	0.5009	0.5093	0.6441	0.7717	0.9171
Peppers	0.2926	0.3422	0.4747	0.5566	0.6606	0.7812	0.8943
Soccer	0.1944	0.2283	0.3243	0.3382	0.4970	0.6353	0.7898
Stdimg	0.3580	0.4148	0.5346	0.6325	0.7557	0.8874	0.9480
Tanaka	0.4033	0.4352	0.5556	0.6178	0.7339	0.8438	0.9264
Woman1	0.2474	0.2988	0.4281	0.4561	0.6164	0.7416	0.8891
Woman2	0.2894	0.3484	0.5021	0.5499	0.6885	0.8365	0.9291
Yatch	0.2983	0.3411	0.4284	0.4403	0.5839	0.7005	0.8382

Table 2.2.2: The probability of having an identical index with the neighbouring block located in the north-side

	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
Airplane	0.4286	0.4586	0.5764	0.6134	0.7062	0.8352	0.9194
Announcer	0.4091	0.4808	0.5976	0.6357	0.7572	0.8389	0.9334
Baboon	0.1083	0.1525	0.2538	0.3059	0.4736	0.6656	0.8277
Boat	0.2797	0.3457	0.4630	0.5185	0.6337	0.7644	0.8907
Cablecar	0.3786	0.4185	0.5346	0.5658	0.6901	0.8109	0.9276
Cornfield	0.3719	0.3923	0.4800	0.4927	0.6133	0.7327	0.8407
Couple	0.3056	0.3459	0.4836	0.5009	0.6523	0.7934	0.8834
Crowd	0.1900	0.2450	0.3694	0.4143	0.5573	0.7307	0.8890
Flower	0.3909	0.4453	0.5726	0.5950	0.7200	0.8194	0.9148
Fruits	0.4046	0.4481	0.6041	0.5945	0.7575	0.8561	0.9035
Girl	0.5354	0.5769	0.7109	0.7019	0.8093	0.8874	0.9445
Hustler	0.5206	0.5650	0.6741	0.6707	0.7674	0.8477	0.9268
Lena	0.4013	0.4573	0.5978	0.6117	0.7339	0.8498	0.9252
Light	0.4820	0.4848	0.6568	0.6189	0.7094	0.8447	0.9122
Man	0.2401	0.3177	0.4483	0.4602	0.6312	0.7782	0.8908
Masuda1	0.5200	0.5668	0.6827	0.6994	0.7965	0.8713	0.9341
Masuda2	0.5769	0.6154	0.7439	0.7630	0.8493	0.9107	0.9475
Model	0.5978	0.6289	0.7318	0.7039	0.8078	0.9006	0.9386
Pens	0.2774	0.3239	0.4539	0.4463	0.6136	0.7643	0.9110
Peppers	0.3855	0.4435	0.5728	0.6413	0.7400	0.8454	0.9298
Soccer	0.2285	0.2727	0.3765	0.3787	0.5530	0.6922	0.8292
Stdimg	0.4805	0.5338	0.6568	0.7340	0.8357	0.9260	0.9657
Tanaka	0.5085	0.5439	0.6736	0.7183	0.8216	0.9005	0.9448
Woman1	0.3137	0.3800	0.5218	0.5465	0.7029	0.8161	0.9195
Woman2	0.3944	0.4594	0.6069	0.6463	0.7669	0.8815	0.9492
Yatch	0.3834	0.4317	0.5362	0.5400	0.6842	0.7998	0.8953

Table 2.2.3: The probability of having an identical index with the neighbouring block located in the east-side

	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
Airplane	0.4957	0.5214	0.6250	0.6551	0.7382	0.8485	0.9196
Announcer	0.4749	0.5379	0.6396	0.6771	0.7833	0.8586	0.9510
Baboon	0.0947	0.1397	0.2408	0.2829	0.4642	0.6664	0.8289
Boat	0.2694	0.3342	0.4601	0.5116	0.6340	0.7603	0.8919
Cablecar	0.3858	0.4317	0.5391	0.5839	0.6930	0.8159	0.9274
Cornfield	0.4245	0.4531	0.5237	0.5456	0.6439	0.7489	0.8545
Couple	0.2583	0.3108	0.4455	0.4634	0.6241	0.7668	0.8731
Crowd	0.1728	0.2275	0.3486	0.4067	0.5378	0.7173	0.8849
Flower	0.3832	0.4297	0.5510	0.5724	0.7178	0.8224	0.9169
Fruits	0.3951	0.4422	0.5925	0.5938	0.7368	0.8407	0.8986
Girl	0.4886	0.5301	0.6698	0.6832	0.8091	0.8910	0.9445
Hustler	0.4695	0.5125	0.6524	0.6527	0.7735	0.8582	0.9237
Lena	0.2763	0.3201	0.4596	0.4795	0.6173	0.7744	0.8890
Light	0.5311	0.5318	0.7081	0.6881	0.7715	0.8864	0.9332
Man	0.2052	0.2709	0.3946	0.4122	0.5966	0.7541	0.8794
Masuda1	0.4969	0.5508	0.6888	0.7165	0.8119	0.8766	0.9327
Masuda2	0.5040	0.5521	0.6848	0.7137	0.8062	0.8768	0.9284
Model	0.4160	0.4594	0.6126	0.5654	0.7306	0.8511	0.9203
Pens	0.2969	0.3557	0.4822	0.4942	0.6380	0.7803	0.9143
Peppers	0.3388	0.3942	0.5247	0.5971	0.7026	0.8178	0.9144
Soccer	0.3199	0.3512	0.4686	0.4894	0.6111	0.7372	0.8576
Stdimg	0.3976	0.4567	0.5722	0.6740	0.8042	0.9136	0.9571
Tanaka	0.4652	0.4991	0.6296	0.6798	0.7887	0.8775	0.9468
Woman1	0.2746	0.3241	0.4567	0.4851	0.6541	0.7755	0.9050
Woman2	0.3593	0.4302	0.5848	0.6264	0.7479	0.8735	0.9434
Yatch	0.4284	0.4781	0.5664	0.5872	0.6977	0.7807	0.8828

Table 2.2.4: The probability of having an identical index with the neighbouring blocks located in the north- and west-side

	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
Airplane	0.6084	0.6487	0.7438	0.7657	0.8478	0.9219	0.9608
Announcer	0.6359	0.7086	0.8041	0.8307	0.9021	0.9441	0.9752
Baboon	0.1752	0.2492	0.3878	0.4663	0.6713	0.8346	0.9187
Boat	0.4044	0.4900	0.6199	0.6684	0.7804	0.8836	0.9539
Cablecar	0.4923	0.5504	0.6690	0.7087	0.8179	0.9078	0.9644
Cornfield	0.4907	0.5301	0.6126	0.6446	0.7608	0.8678	0.9352
Couple	0.4697	0.5355	0.6790	0.7059	0.8301	0.9163	0.9563
Crowd	0.2935	0.3736	0.5191	0.5791	0.7160	0.8541	0.9478
Flower	0.5347	0.5979	0.7232	0.7460	0.8405	0.9096	0.9587
Fruits	0.5426	0.6135	0.7556	0.7542	0.8732	0.9289	0.9500
Girl	0.6996	0.7454	0.8464	0.8481	0.9093	0.9522	0.9728
Hustler	0.6865	0.7305	0.8279	0.8292	0.8992	0.9425	0.9712
Lena	0.4927	0.5607	0.6923	0.7094	0.8165	0.9067	0.9541
Light	0.6516	0.6812	0.8144	0.8030	0.8749	0.9436	0.9660
Man	0.3550	0.4504	0.6001	0.6258	0.7802	0.8860	0.9475
Masuda1	0.6867	0.7408	0.8363	0.8499	0.9165	0.9520	0.9713
Masuda2	0.6932	0.7372	0.8384	0.8513	0.9173	0.9574	0.9728
Model	0.7079	0.7484	0.8382	0.8221	0.9015	0.9535	0.9712
Pens	0.4116	0.4928	0.6517	0.6566	0.8037	0.9088	0.9693
Peppers	0.5155	0.5850	0.7097	0.7598	0.8389	0.9149	0.9643
Soccer	0.3951	0.4530	0.5920	0.6131	0.7537	0.8672	0.9423
Stdimg	0.6007	0.6553	0.7666	0.8285	0.9037	0.9604	0.9782
Tanaka	0.6420	0.6949	0.8152	0.8421	0.9118	0.9549	0.9720
Woman1	0.4414	0.5227	0.6678	0.6904	0.8239	0.9035	0.9574
Woman2	0.5509	0.6263	0.7717	0.7950	0.8827	0.9448	0.9730
Yatch	0.5581	0.6216	0.7191	0.7415	0.8440	0.9142	0.9593

Table 2.2.5: The probability of having an identical index with the neighbouring blocks located in the north-, west-, and north-west-side

	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
Airplane	0.6323	0.6788	0.7646	0.7921	0.8658	0.9307	0.9635
Announcer	0.6662	0.7371	0.8253	0.8489	0.9115	0.9481	0.9757
Baboon	0.2094	0.2979	0.4506	0.5463	0.7451	0.8762	0.9375
Boat	0.4412	0.5288	0.6560	0.7086	0.8150	0.9023	0.9575
Cablecar	0.5228	0.5864	0.6981	0.7463	0.8450	0.9215	0.9680
Cornfield	0.5074	0.5539	0.6394	0.6833	0.7990	0.8956	0.9469
Couple	0.4963	0.5689	0.7032	0.7374	0.8505	0.9257	0.9596
Crowd	0.3256	0.4095	0.5594	0.6253	0.7563	0.8757	0.9519
Flower	0.5819	0.6449	0.7548	0.7756	0.8598	0.9194	0.9606
Fruits	0.5862	0.6617	0.7858	0.7973	0.8931	0.9384	0.9557
Girl	0.7320	0.7758	0.8609	0.8666	0.9205	0.9557	0.9734
Hustler	0.7085	0.7542	0.8419	0.8444	0.9075	0.9469	0.9722
Lena	0.5293	0.6000	0.7203	0.7437	0.8418	0.9186	0.9582
Light	0.6768	0.7176	0.8345	0.8319	0.8934	0.9483	0.9671
Man	0.3966	0.5041	0.6439	0.6810	0.8117	0.9007	0.9524
Masuda1	0.7228	0.7732	0.8549	0.8672	0.9249	0.9553	0.9722
Masuda2	0.7314	0.7757	0.8602	0.8738	0.9273	0.9606	0.9730
Model	0.7386	0.7813	0.8530	0.8479	0.9152	0.9601	0.9734
Pens	0.5272	0.6108	0.7444	0.7573	0.8629	0.9325	0.9720
Peppers	0.5605	0.6265	0.7428	0.7874	0.8571	0.9225	0.9654
Soccer	0.4227	0.4909	0.6263	0.6546	0.7880	0.8867	0.9467
Stdimg	0.6349	0.6922	0.7957	0.8495	0.9135	0.9624	0.9783
Tanaka	0.6806	0.7400	0.8382	0.8646	0.9218	0.9578	0.9730
Woman1	0.4960	0.5838	0.7164	0.7426	0.8536	0.9170	0.9611
Woman2	0.5999	0.6732	0.8009	0.8235	0.8969	0.9492	0.9735
Yatch	0.5818	0.6511	0.7387	0.7653	0.8588	0.9225	0.9620

Table 2.2.6: The probability of having an identical index with the neighbouring blocks located in the north-, west-, north-west-, and north-east-side

	7 bpv	6 bpv	5 bpv	4 bpv	3 bpv	2 bpv	1 bpv
Airplane	0.6635	0.7169	0.7983	0.8261	0.8960	0.9512	0.9739
Announcer	0.7010	0.7740	0.8547	0.8760	0.9284	0.9586	0.9778
Baboon	0.2488	0.3508	0.5128	0.6257	0.8140	0.9137	0.9545
Boat	0.4813	0.5733	0.6982	0.7533	0.8520	0.9264	0.9665
Cablecar	0.5603	0.6317	0.7388	0.7887	0.8810	0.9417	0.9727
Cornfield	0.5225	0.5728	0.6609	0.7131	0.8304	0.9194	0.9582
Couple	0.5283	0.6071	0.7372	0.7749	0.8810	0.9427	0.9680
Crowd	0.3752	0.4764	0.6243	0.6915	0.8189	0.9164	0.9671
Flower	0.6327	0.7011	0.8036	0.8280	0.8999	0.9490	0.9726
Fruits	0.6431	0.7210	0.8337	0.8462	0.9278	0.9563	0.9658
Girl	0.7825	0.8208	0.8924	0.9014	0.9424	0.9671	0.9773
Hustler	0.7410	0.7902	0.8658	0.8736	0.9266	0.9590	0.9776
Lena	0.5985	0.6735	0.7828	0.8123	0.8945	0.9533	0.9759
Light	0.7195	0.7700	0.8670	0.8646	0.9212	0.9620	0.9723
Man	0.4511	0.5696	0.7101	0.7470	0.8667	0.9356	0.9679
Masuda1	0.7608	0.8080	0.8802	0.8943	0.9417	0.9639	0.9766
Masuda2	0.7716	0.8160	0.8905	0.9017	0.9463	0.9701	0.9788
Model	0.7737	0.8164	0.8801	0.8828	0.9383	0.9692	0.9781
Pens	0.5504	0.6408	0.7661	0.7824	0.8803	0.9420	0.9743
Peppers	0.6168	0.6876	0.7937	0.8331	0.8947	0.9491	0.9785
Soccer	0.4492	0.5237	0.6564	0.6909	0.8234	0.9097	0.9563
Stdimg	0.6922	0.7557	0.8525	0.8937	0.9461	0.9722	0.9815
Tanaka	0.7373	0.8004	0.8852	0.9050	0.9463	0.9712	0.9781
Woman1	0.5561	0.6583	0.7816	0.8086	0.9060	0.9526	0.9744
Woman2	0.6671	0.7386	0.8522	0.8741	0.9318	0.9678	0.9805
Yatch	0.6153	0.6840	0.7674	0.7932	0.8810	0.9377	0.9691

Appendix B:The Effect of Block Size on the Performance of Index Compression

1. Results of IC-VQ (III) for the test image "*Lena*"

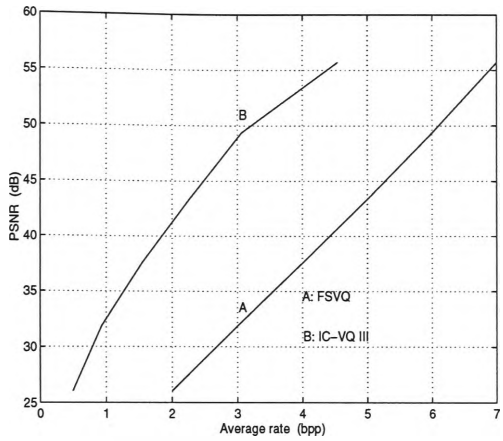

Figure B.1: Results for block size 1x1 (*Lena*)

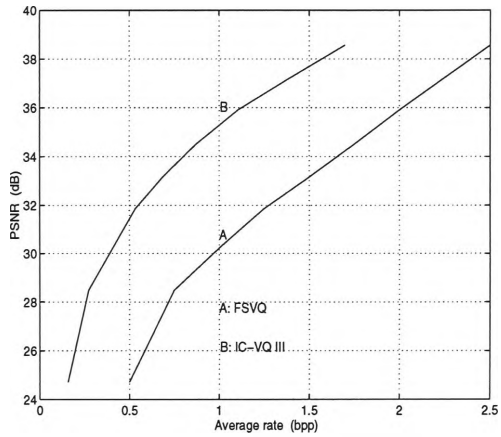

Figure B.2: Results for block size 2x2 (*Lena*)

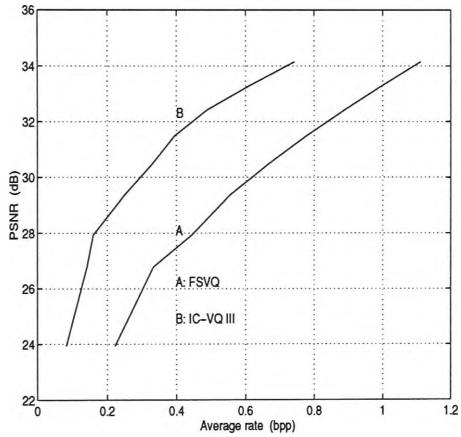

Figure B.3: Results for block size 3x3 (*Lena*)

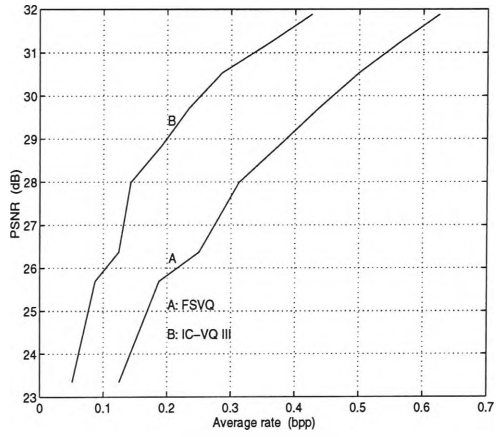

Figure B.4: Results for block size 4x4 (*Lena*)

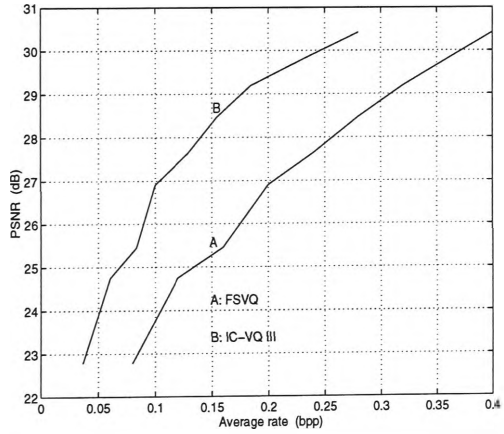

Figure B.4: Results for block size 5x5 (*Lena*)

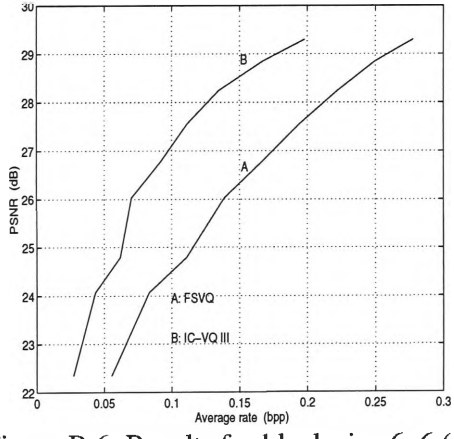

Figure B.6: Results for block size 6x6 (*Lena*)

2. Results of IC-VQ (III) for the test image "*Airplane*"

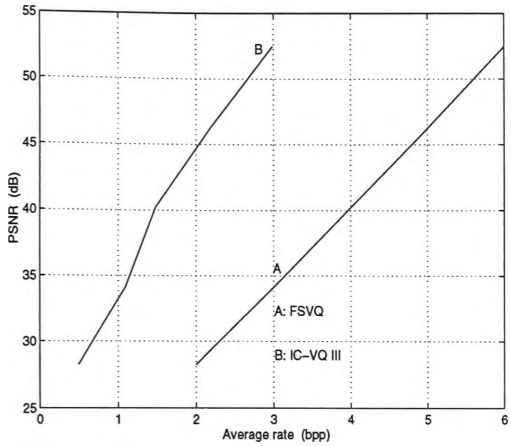

Figure B.7: Results for block size 1x1 (*Airplane*)

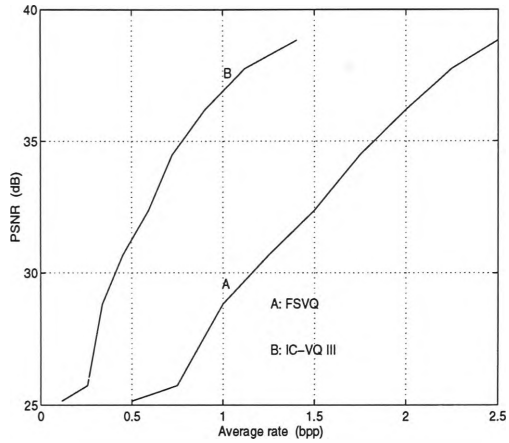

Figure B.8: Results for block size 2x2 (*Airplane*)

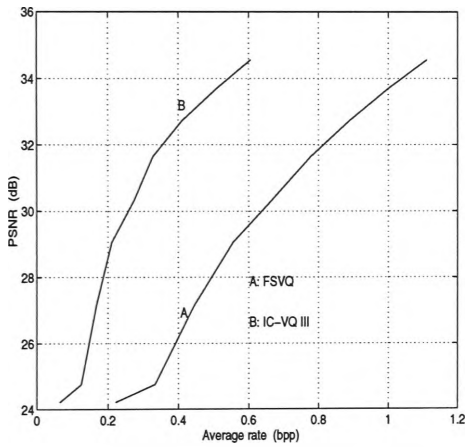

Figure B.9: Results for block size 3x3 (*Airplane*)

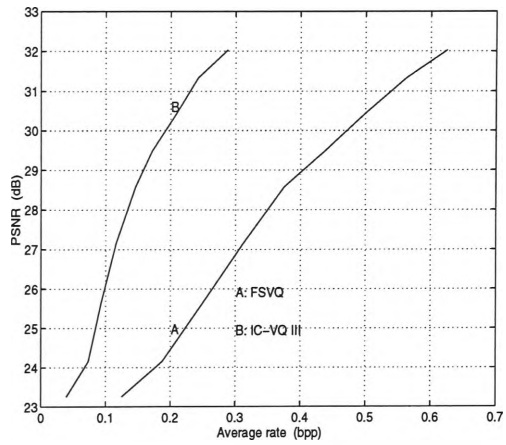

Figure B.10: Results for block size 4x4 (*Airplane*)

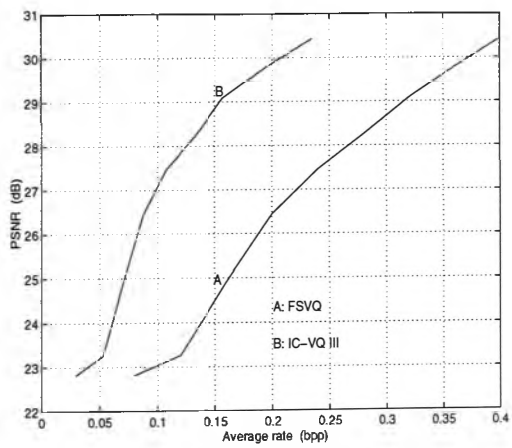

Figure B.11: Results for block size 5x5 (*Airplane*)

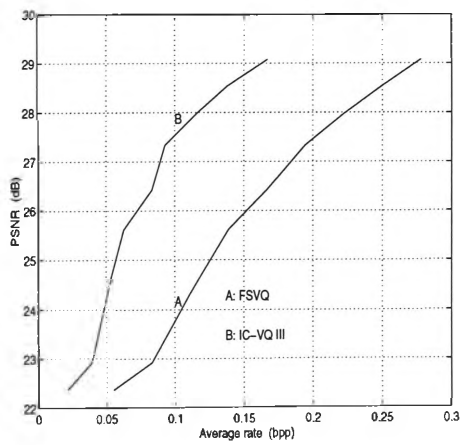

Figure B.12: Results for block size 6x6 (*Airplane*)

3. Results of IC-VQ (III) for the test image "*Stdimg*"

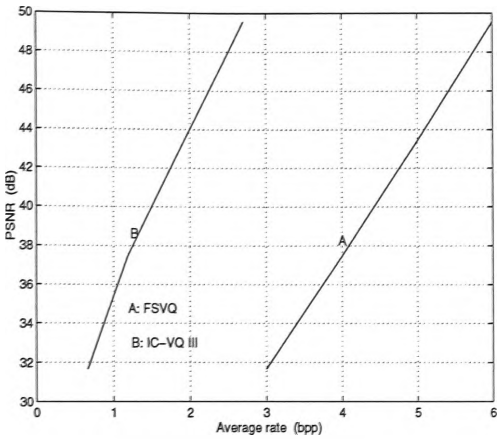

Figure B.13: Results for block size 1x1 (*Stdimg*)

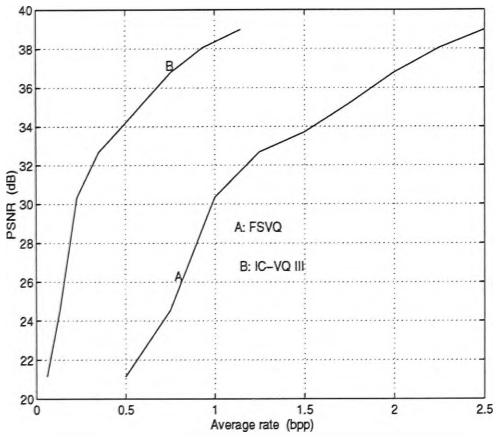

Figure B.14: Results for block size 2x2 (*Stdimg*)

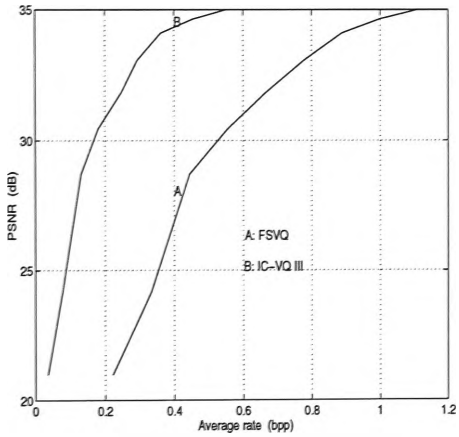

Figure B.15: Results for block size 3x3 (*Stdimg*)

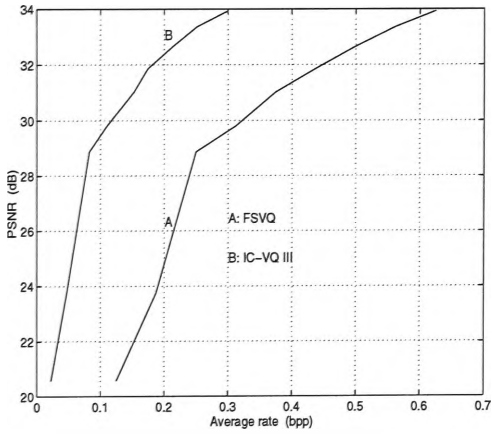

Figure B.16: Results for block size 4x4 (*Stdimg*)

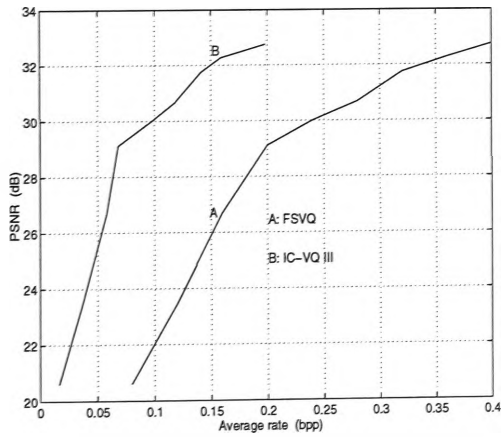

Figure B.17: Results for block size 5x5 (*Stdimg*)

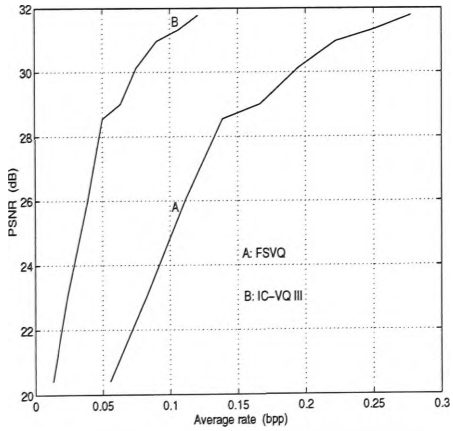

Figure B.18: Results for block size 6x6 (*Stdimg*)

4. Results of IC-VQ (III) for the test image "*Baboon*"

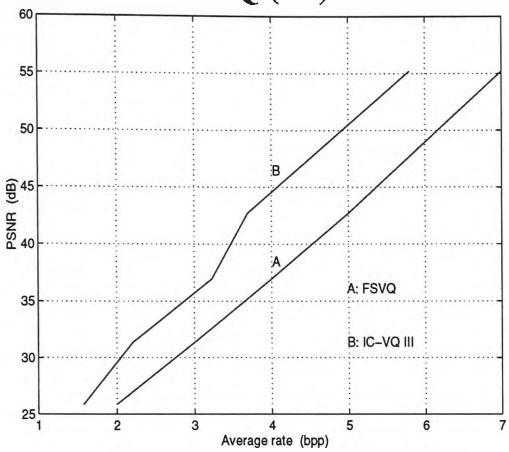

Figure B.19: Results for block size 1x1 (*Baboon*)

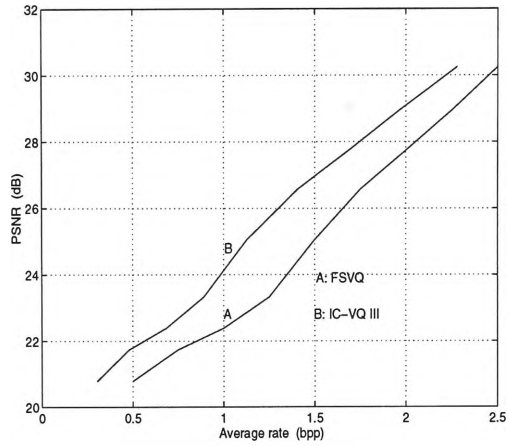

Figure B.20: Results for block size 2x2 (*Baboon*)

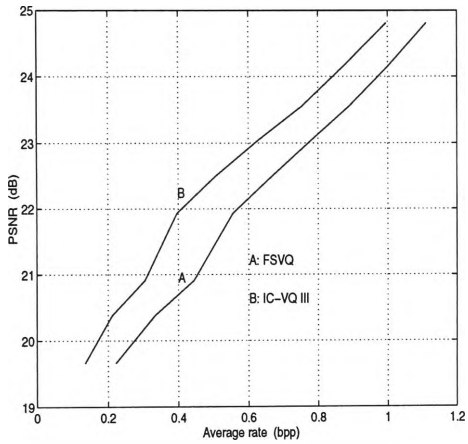

Figure B.21: Results for block size 3x3 (*Baboon*)

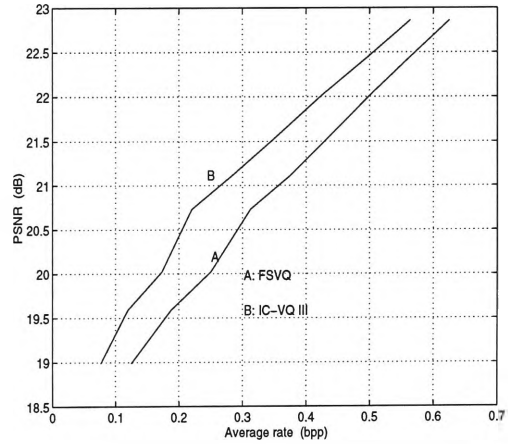

Figure B.22: Results for block size 4x4 (*Baboon*)

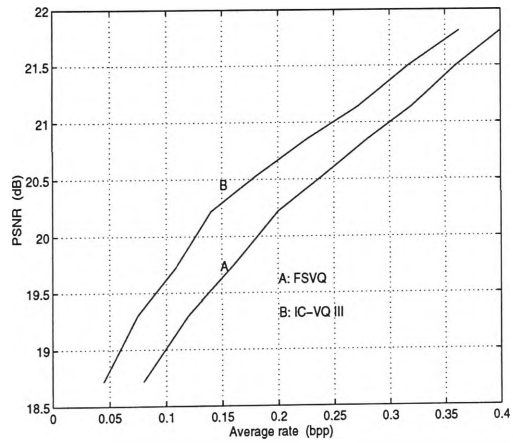

Figure B.23: Results for block size 5x5 (*Baboon*)

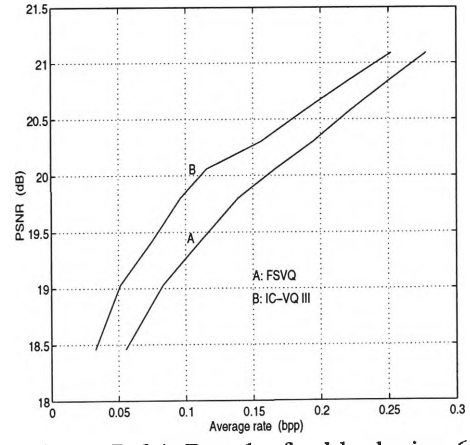

Figure B.24: Results for block size 6x6 (*Baboon*)